

# **Réseaux et Protocoles**

**NFP 104**

**Gérard Florin , Stéphane Natkin**

**Conservatoire National des Arts et Métiers**



# **PLAN DU COURS**

## **Les couches basses**

### **INTRODUCTION NOTION GÉNÉRALES**

#### **1. NIVEAU PHYSIQUE**

##### **1.1. Transmission sur un canal**

**Transmission en bande limitée**

**Transmission en présence de bruit**

**Détection et correction des erreurs**

**Représentation des signaux**

##### **1.2. Éléments de technologie**

Les contrôleurs

Les interfaces standards

Les modems

Les voies de communication

Les réseaux au niveau physique

#### **2. NIVEAU LIAISON**

##### **2.1 LIAISON POINT A POINT**

. Problèmes généraux de réalisation des protocoles de liaison

Introduction

Délimitation des trames

Protocoles de complexité croissante

. Exemples

Protocoles à trames de bits type HDLC-LAPB

##### **2.2 LIAISON DANS LES RÉSEAUX LOCAUX**

. Introduction

. Protocoles en compétition: exemple Ethernet

. Protocoles en coopération:

#### **3. NIVEAU RÉSEAU**

##### **3.1 PROBLÈMES DE RÉALISATION DE LA COUCHE RÉSEAU**

. Circuits virtuels et datagrammes

. Adressage

. Routage

. Contrôle de congestion

##### **3.2 EXEMPLES DE PROTOCOLE :**

. IP

. ATM

#### **4. NIVEAU TRANSPORT**

##### **4.1 PROBLÈMES DE RÉALISATION DE LA COUCHE TRANSPORT**

. Ouverture et fermeture de connexion

. Régulation de flux

. Détection et correction d'erreur

##### **4.2 EXEMPLE DE PROTOCOLE**

. TCP

## **BIBLIOGRAPHIE**

**Andrew Tanenbaum : "Réseaux", Pearson Education, 4<sup>ième</sup> édition 2003.  
Traduction en français de l'ouvrage Computer Networks**

**Claude Servin : Réseaux et Telecom , Dunod, Paris 2003**

**James F Kurose, Keith W Ross Computer Networking. Addison Wesley  
2001**

Toutain. L : Réseaux locaux et Internet Hermès

Stevens W.R.: " TCP/IP Illustrated", Addison Wesley 1993.

Boisseau, Demange, Munier, “ Réseaux Haut débits”, Eyrolles, ed 1993

Ed Krol, “The whole Internet”, O’Reilly, 1992

Comer, “Internetworking with TCP/IP”, vol 1,2,3, Prentice Hall, 1992

Bouyer, G. "Les réseaux synchrones étendus PDH et SDH", Hermès

**Les normes OSI, UIT,**

**Les RFC INTERNET,**

**Les groupes de travail, forums etc**

**Les serveurs WEB constructeurs, universitaires, ...**

# **PREMIER CHAPITRE**

## **Notions Générales**

# **I**

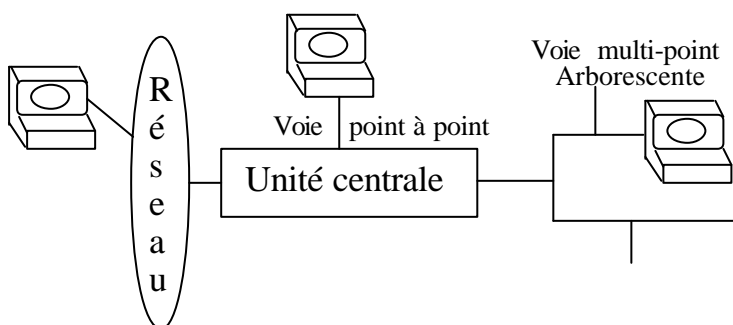
## **Évolution des réseaux, Perspectives**

# I.1 Le télétraitement ("Teleprocessing")

## Objectif :

S'affranchir des **contraintes géographiques** de localisation des systèmes informatiques.

- Au départ **rentabiliser** des unités centrales coûteuses.
- Installer les terminaux près des utilisateurs.
- Mettre **en place une organisation des moyens** informatiques qui s'adapte à l'approche centralisée.



# Compléments: télétraitement

## **Privilège une vision centralisée des traitements.**

- Un système informatique important supporte une application

- Celle-ci est accédée à distance au moyen de postes de travail plus ou moins intelligents mais dont le rôle majeur est celui d'un terminal (la saisie).

Exemples : terminaux caractères, Minitels, PC, terminaux X.

## **L'acheminement des informations**

S'effectue par différents moyens:

. Liaisons spécialisées point à point.

. Voies multipoints : organisation arborescente.

. Utilisation de réseaux (ramassage des données)

=> Réseau téléphonique.

=> Réseaux spécialisés de transmission de données.

## **Applications types**

Banque, réservation, ...

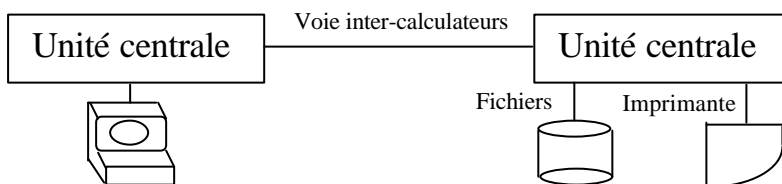


## I.2 Évolution vers l'interconnexion en réseau des processeurs

### Gestion des ressources

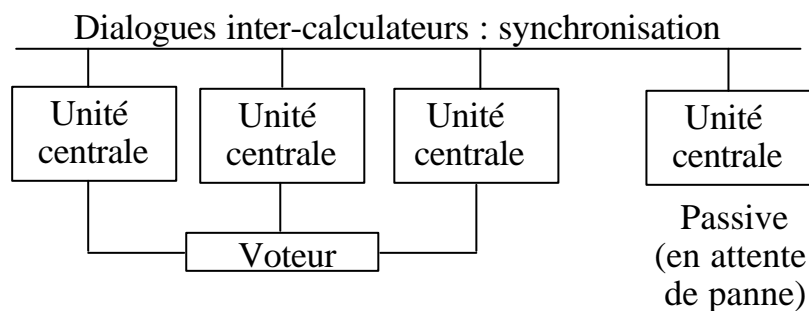
**Objectif** : Satisfaire des besoins d'optimisation des ressources, d'accès à des ressources distantes rares.

Exemples : périphériques spéciaux/rapides, puissance de calcul (architectures multiprocesseurs), logiciels.



### La tolérance aux pannes

**Objectif** : Permettre à des applications sensibles de continuer de fonctionner en présence de pannes.



## Évolutions vers les réseaux

### Partage des ressources ("Resource Sharing")

1) **Évolution 1:** Mise en place d'architectures avec des processeurs spécialisés reliés à faible distance

- **Processeur central - Canaux d'entrées sorties (1960)**

- **Processeur central - Processeurs spécialisés**

(fin années 1960)

. Dans la gestion des voies réseaux

(Frontaux "Front-End")

. Dans la gestion des disques, ...

- **Architectures Multiprocesseurs**

. Plusieurs processeurs connectés sur le même bus coopèrent au moyen du partage d'une mémoire commune.

2) **Évolution 2 : Vers la banalisation des processeurs et des ressources gérées** (fin années 1960)

. Interconnexion dans des réseaux généraux de processeurs reliés à des distances quelconques.

### Tolérance aux pannes ("Fault Tolerance")

Haute disponibilité (informatique de gestion)

Haute sécurité (informatique industrielle)

Projet Appollo (1964-1969)

**Applications avec redondance de moyens matériels .**

. Dispositifs matériels spéciaux: commutateurs, voteurs

. Nécessité d'architectures de communications pour la synchronisation des applications redondantes.

# **Les réseaux généraux d'ordinateurs ("WAN Wide Area Networks")**

## **Expérimentation du réseau D-ARPA**

("Defense Advanced Research Project Agency")

Ensemble de travaux universités industries sur contrat à partir de 1969

Développement des principes essentiels des réseaux informatiques

- Protocoles de communications couches basses
- Développement d'un ensemble de protocoles d'application (courant 1970)
  - . Sessions à distance
  - . Transport de fichiers
  - . Messagerie

## **Annnonce des architectures constructeurs**

- IBM SNA (1974) ("System Network Architecture")

## **Normalisation des réseaux publics**

- X25 (1974)

## **Notions relatives aux réseaux généraux**

### **Réseau général**

Ensemble des systèmes informatiques autonomes capables d'échanger des informations en univers hétérogène.

#### **Autonome**

- Mémoire propre à chacun des sites.
- Pas de synchronisation matérielle puissante.
  - . pas de mémoire partagée
  - . pas de synchronisation d'horloge

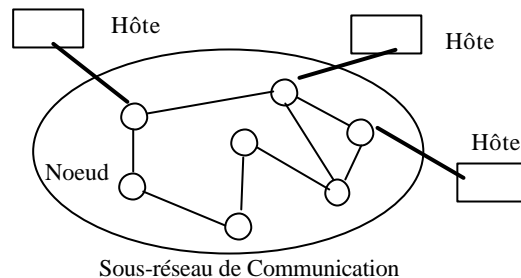
#### **Echange**

- Communications en mode message asynchrone.  
"Asynchronous message passing"
- Débit plutôt faible (< 2Mb/s idée de faible couplage).
- Sur des distances quelconques.

#### **Hétérogénéité**

Capable de faire fonctionner ensemble des systèmes d'origine différentes.

# Terminologie des réseaux généraux



## Ordinateurs Hôtes ("Hosts Computers")

Les systèmes informatiques interconnectés.

## Sous-réseau de communication ("Communication Subnet")

Le moyen de communication des hôtes.

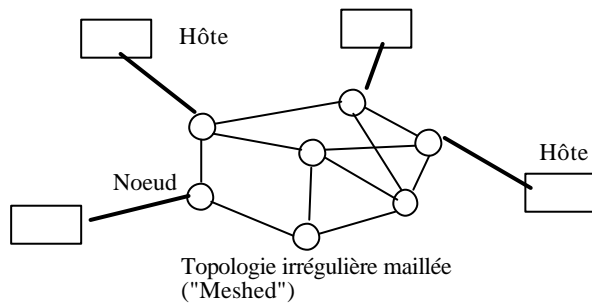
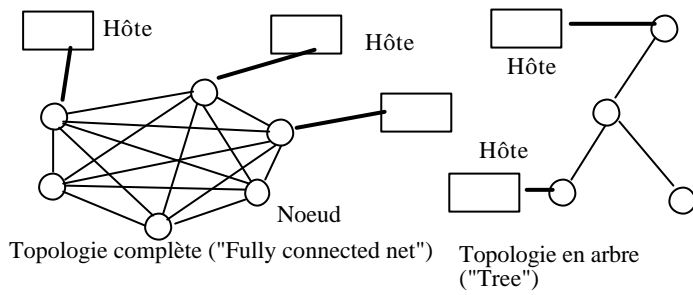
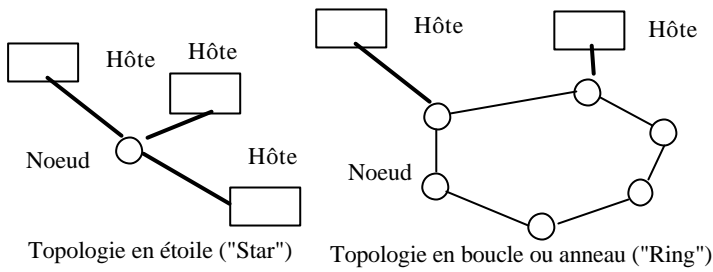
**- Des calculateurs spécialisés (ou non) dans la commutation.**

- . Commutateurs de paquets  
noeuds de commutation, routeurs
- . "Packet Switches", "Nodes", "Routers"
- . IMP "Interface Message Processor"

**- Des voies physiques de communication.**

- . Circuits , voies, canaux - "Channels"

# Topologies des réseaux



## **Le développement des réseaux informatiques vers les communications interpersonnelles**

- Diffusion de plus en plus large des applications préexistantes des réseaux (téléinformatique).
- Convergence de plusieurs tendances dans les années 1970: **La télématique (1981)**

### **Réseaux généraux**

Les réseaux d'ordinateurs sont des supports fiables de transport de messages textuels.

Résultat de l'expérience ARPA.  
Développement des messageries.

### **Techniques de commutation**

Les techniques de construction des autocommutateurs évoluent vers les techniques numériques et la commutation temporelle synchrone.

Premier commutateur temporel 1975

### **Concrétisation: le RNIS Réseau Numérique à Intégration de Service (début des années 1980)**

Intégration sur le même support de transmissions voix, données et images (faiblement animées)

Débits de base 64 kilobits/seconde, 2 mégabits/seconde

## I.3 Les réseaux locaux et les systèmes répartis ("LAN Local Area Networks")

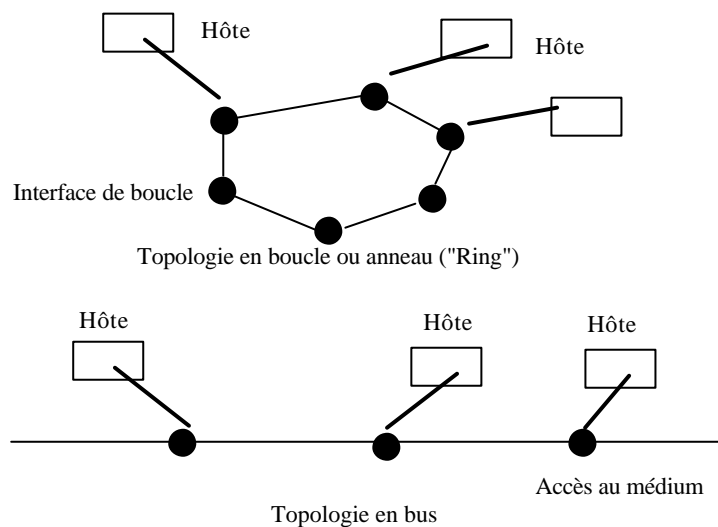
Expérimentation du réseau ethernet à partir de 1974  
Diffusion à grande échelle à partir de 1980

### Architectures de réseaux locaux (fortement couplés)

Communications en mode message asynchrone.

A débit élevé  $> 2$  Mégabits/seconde.

Sur des distances faibles.



### Topologies de réseaux locaux



## **Développement des architectures de réseaux d'entreprise**

Interconnexion des postes de travail

- accès aux systèmes informatiques d'entreprise
- accès aux outils réseaux (messagerie, ...)

Développement des ateliers logiciels.

Développement des réseaux locaux industriels.

## **Les systèmes répartis ("Distributed System")**

### **Notion de système réparti ou distribué (à partir de 1980)**

Approche d'homogénéisation des systèmes

Système d'exploitation commun à tous les sites d'un réseau permettant d'offrir un service réseau équivalent à celui d'un processeur unique.

Résolution des problèmes de désignation.  
Exemple : localisation des fichiers

Résolution de problèmes de gestion de ressources.  
Exemple : sélection d'un processeur pour  
exécution d'une tâche.

Résolution de problèmes de synchronisation.  
Exemple : contrôle d'exécution répartie

Exemples: Mach, Chorus

### **Développement de l'algorithmique parallèle et distribuée**

Structuration des applications en univers réparti.

Parallélisation des algorithmes.

## **I.4 Les développements en cours**

### **Architecture des machines et des réseaux**

Le parallélisme dans tous les systèmes informatiques.

Le parallélisme massif.

Les réseaux haut débits sur fibre optique.

### **Les domaines applicatifs concernés par le parallélisme et les réseaux : tous**

#### **L'informatique industrielle**

Les systèmes répartis de contrôle de procédés

#### **L'informatique scientifique**

La parallélisation

#### **L'informatique d'entreprise**

Les architectures client-serveur

#### **Les systèmes informatiques répartis**

Les systèmes d'objets répartis

#### **L'intelligence artificielle**

Les réseaux de neurones

Les systèmes multi-agents

#### **Le multi média et la télévision interactive**

## **II**

### **Interconnexion des systèmes ouverts:**

#### **Modèle de référence**

## **II.1 Introduction aux systèmes ouverts**

- **Un modèle de référence d'architecture de systèmes ouverts (architectures de réseaux)**

**Pour quoi faire?**

**Pour permettre la construction rationnelle d'applications réparties**

- . Exploitation des possibilités réseaux,
- . Parallélisme,
- . Extensibilité,
- . Tolérance aux pannes.

**Pour régler des problèmes d'incompatibilité entre différents choix techniques.**

**=> Notion d'ouverture**

# Les approches de standardisation d'architectures

- **Hétérogénéité / Compétition entre solutions**
  - . à l'intérieur d'un même constructeur,
  - . d'un même organisme important.
  - . d'une même entreprise,

=> On aboutit à une **variété importante de protocoles incompatibles** engendrant des surcoûts considérables.

Les architectures de référence constructeurs:

- **SNA** ("System Network Architecture") IBM
- **DNA** ("Digital Network Architecture") Digital(Decnet)
- **DSA** ("Distributed System Architecture") BULL

Les organismes imposant des standards

- **OSI** - Organisation des Standards Internationaux.
- **DARPANET** ("Defense Advanced Research Project Agency Network") Défense Américaine -> INTERNET

- **Compétition entre les différentes propositions**
  - Solutions de toutes origines => Normalisation.
  - => Compétition entre les différentes propositions

Compétiteurs majeurs: OSI, SNA, INTERNET.

# Environnements de systèmes ouverts

("OSE Open System Environment")

## Objectifs

- Assurer que des systèmes informatiques et leur logiciels puissent être intégrés dans des ensembles coopérants de grande dimension sans supporter des coûts importants.

## Moyens

### - **L'interopérabilité** ("Interoperability")

Signifie que des systèmes ouverts interconnectés peuvent échanger des informations significatives.

=> Non seulement la connexion physique est réalisée mais également pour tous les protocoles employés il y a interfonctionnement.

### - **La portabilité des applications** ("Portability")

Signifie que le même logiciel peut être exécuté sur une grande variété de machines.

=> On peut ainsi différencier les fournisseurs.

## - L'extensibilité ("Scalability")

Signifie que le système ouvert peut supporter les extensions de configuration.

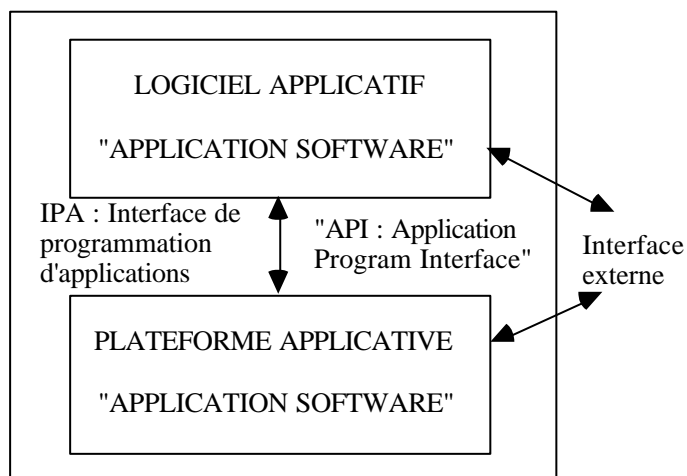
=> Par son développement le système ouvert peut accompagner l'extension des applications en leurs assurant selon la demande des performances acceptables).

## - L'intégration ("Integration")

Signifie que les applications ont une interface bien définie et correctement utilisée.

=> Les applications peuvent être assemblées pour former un système complexe.

Notion d'interface applicative: API





## II. 2 Organisation d'une architecture de réseaux

### - Organisation en couches ou niveaux ("Layers")

Pour modulariser la conception

Pour structurer les différentes fonctions à réaliser

**=> Éviter les approches "fourre-tout"**

On situe dans une hiérarchie de couches (une pile ou "stack") les différentes fonctions à réaliser.

On définit la façon dont se comporte chaque niveau:

- En termes du service rendu au niveau supérieur.
- Par la façon de dialoguer entre implantations de niveaux analogues.

### - Organisation orientée objet

On évite la définition d'une pile (hiérarchisée) de niveaux en mettant tous les modules sur un même plan.

En cours de développement.

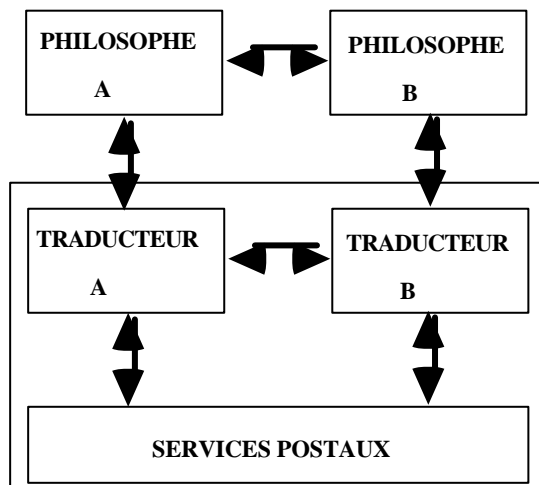
# Services et protocoles

("Service", "Protocol")

## Exemple introductif

Deux philosophes qui ne parlent pas la même langue souhaitent mener un débat philosophique par la poste.

- Ils produisent des textes.
- Ils utilisent les services de traducteurs.
- Les textes circulent par la poste.



## Notion de service (flèches verticales)

### Interface du service de traduction (requêtes ou primitives)

- Pourriez vous envoyer un texte à mon ami le philosophe B qui habite a telle adresse.
- Oui c'est possible.
- Non j'ai trop de travail.
- Un texte pour vous est arrivé de B.

### Interface du service postal

- Mettre une lettre à la poste
- Effectuer un envoi recommandé (qualité du service)
  - . Guichet surchargé (file d'attente)
  - . Poste fermée.

## Notion de protocole (flèches horizontales)

### Protocole entre philosophes

- Cher et estimé collègue.
- Thèse, antithèse, synthèse.

### Protocole entre traducteurs

- Votre cinquième paragraphe du dernier texte était incompréhensible.
  - Dans une langue "pivot" (anglais) : Que pensez vous d'utiliser pour le prochain envoi l'allemand afin de ne pas perdre la main.

# Architecture de communication

## Présentation formelle des notions

### Architecture de réseau ("Network Architecture")

Spécification d'un ensemble de fonctions découpées en niveaux hiérarchisés

### Couches ou niveaux ("Layers")

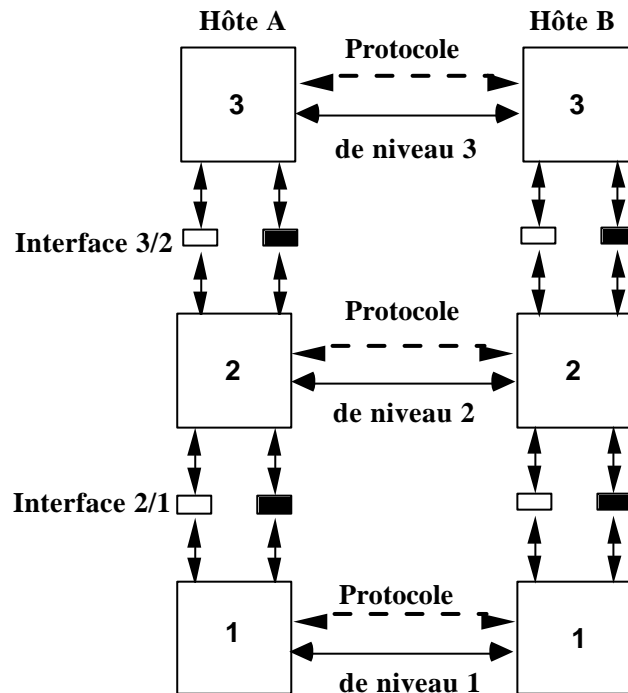
Une couche N est définie par:

- **Une interface de service**  
(des primitives d'accès au service).
- **Un protocole de communication**  
(entre un niveau N et un niveau N distant).

### Services et protocoles

- Le niveau N communique avec le niveau N+1 auquel **il fournit un service.**
- Le niveau N communique avec le niveau N-1 auquel **il demande un service**
- Les services rendus servent à établir finalement **un dialogue** (protocole) entre **deux niveaux N appariés.**
- Le niveau **le plus bas** est celui de la **communication effective** sur une voie physique de bits d'information.

# Représentation des Niveaux, Protocoles, Services



Sur le schéma: distinction des flots de contrôle et flots de données.  
Selon les choix de conception ces flots:

- circulent sur le même canal physique /voie logique
- circulent sur des canaux physiques /voies logiques différentes.

## Définition des notions liées au service

### "Primitive" de service

. **Spécification d'une fonction** précise pouvant être activée dans un niveau par le niveau supérieur.

Exemple : demande de connexion, demande de transfert de données....

. **Caractéristiques associées** : les paramètres

- type de l'opération
- adresse du destinataire
- spécification de qualité de service attendu
- données usagers

### Service

**Ensemble de primitives** offertes par un niveau donné à un niveau supérieur associées à leurs règles de bon usage.

- **Profil d'appel des primitives**

Analogie de la signature,

- **Contraintes d'enchaînement.**

Dans un certain état d'un niveau certaines primitives sont utilisables et d'autres pas.

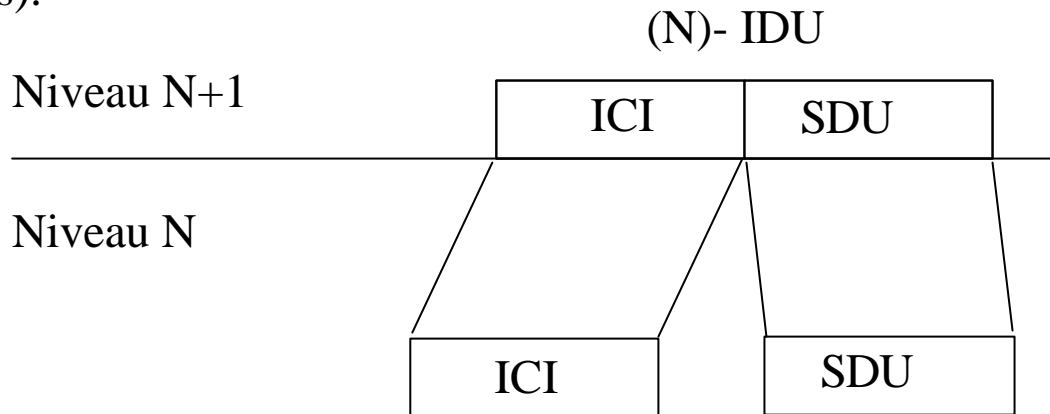
L'interprétation d'une même requête peut être différente selon l'état.

### Notion d'entité de service

**Instance** d'un ensemble de primitives permettant la réalisation effective du service (analogie de l'instanciation de classe en approche objet).

## Unité de données d'interface ("IDU Interface Data Unit")

Ce sont les **objets échangés entre les niveaux** lors de l'émission d'une primitive de service (composés de deux parties).



### - Unités de données de service

("SDU (Service Data Unit)")

. La partie que l'utilisateur souhaite **transmettre effectivement** à l'utilisateur distant.

### - Informations de contrôle de l'échange

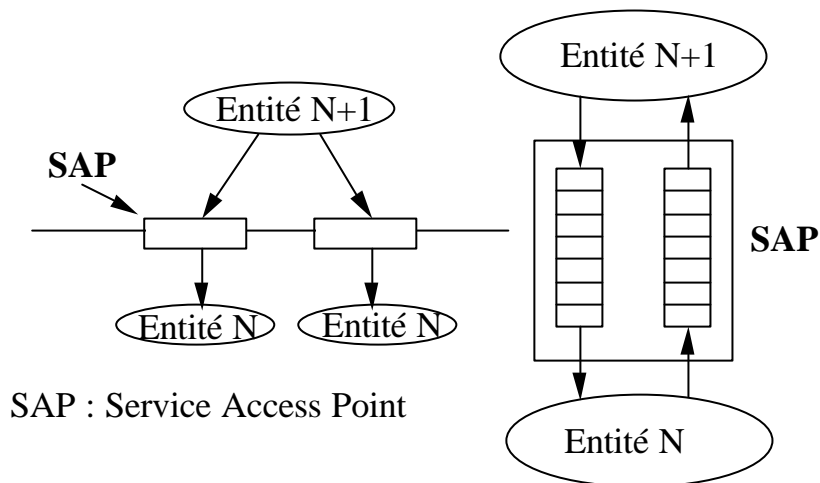
("ICI "Interface Control Information")

. C'est l'ensemble des informations de contrôle qui permettent au niveau inférieur de traiter correctement la requête.

#### 1 Le type de la requête, l'adresse destinataire, ...

2 **Autres informations** dans le dialogue entre niveaux (par exemple pour réguler le flux d'informations sur l'interface de service). **La normalisation ne spécifie pas comment sont échangés les SDU.**

## Points d'Accès de Service ("SAP Service Access Point")



SAP : Service Access Point

Réalisation d'un SAP  
en schéma producteur-consommateur

. **Guichet** permettant à une entité de demander un service à une autre.

Sa réalisation dépend des choix d'implantation du logiciel réseau par le fournisseur (exemple en mode producteur consommateur)

. Le point d'accès de service est l'élément essentiel de la **désignation** dans les réseaux.

Autres dénominations: **port, porte, "sockets", prises.**



# Définition des notions liées aux protocoles

**Unités de données protocolaires**  
**PDU Protocol Data Unit**  
**(trames, paquets, messages, segments)**

. **Spécification d'un ensemble de données** typé échangé entre deux niveaux appariés.

Exemple : transfert de données...

- . **Caractéristiques associées** : les paramètres
  - type de l'opération
  - adresse du destinataire
  - informations auxiliaires transportées
  - données usagers

## Protocole

**Définition des unités de données échangées** par un niveau avec le niveau apparié et **de leurs contraintes d'enchaînement.**

- Dans un état d'un niveau certains messages sont interprétables et d'autres pas.

- L'interprétation d'un même message peut être différente selon l'état.

=> **Indéterminisme** des applications réseaux.

# Unités de données de protocole ("PDU Protocol Data Unit")

L'ensemble des objets échangés entre niveaux appariés.

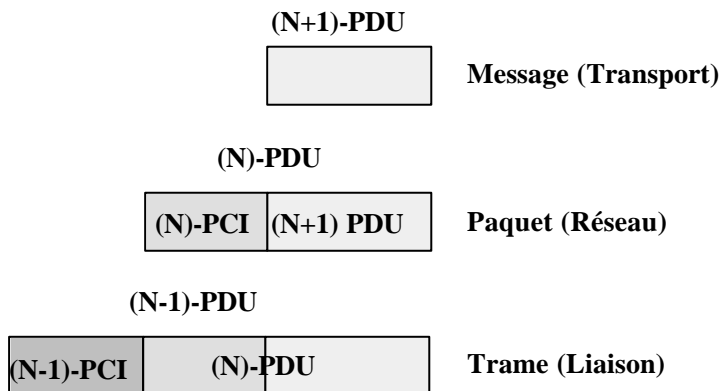
Un (N)-PDU est composé d'un (N+1)-PDU et d'une information de contrôle (N)-PCI.

## - Informations de contrôle protocolaire ("PCI "Protocol Control Information")

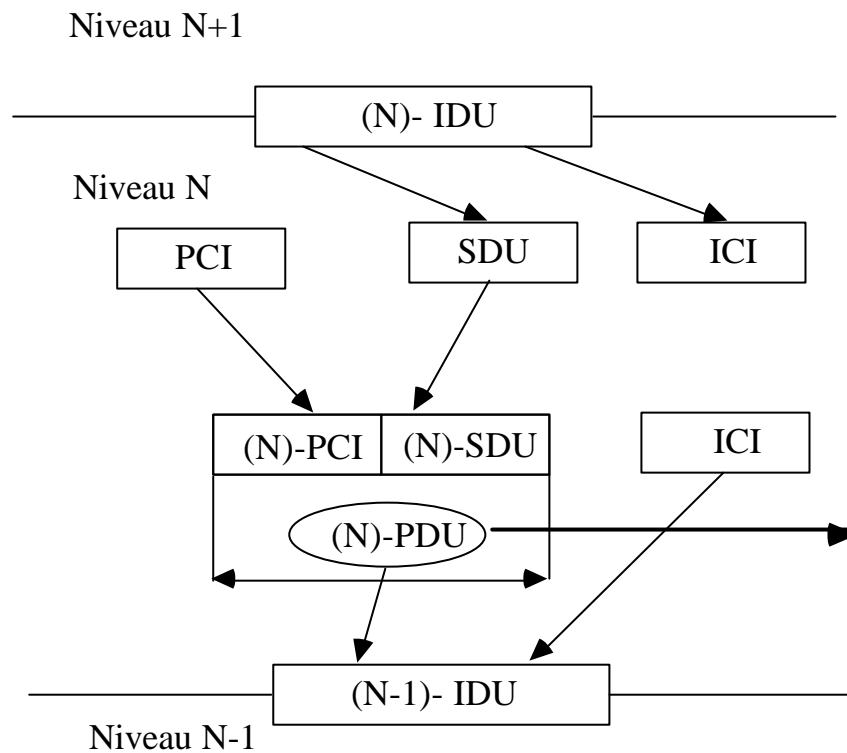
. Ensemble des informations de contrôle de l'échange entre niveaux.

Exemples :    adresses émetteur et destinataire  
                  type  
                  version de protocole utilisée.

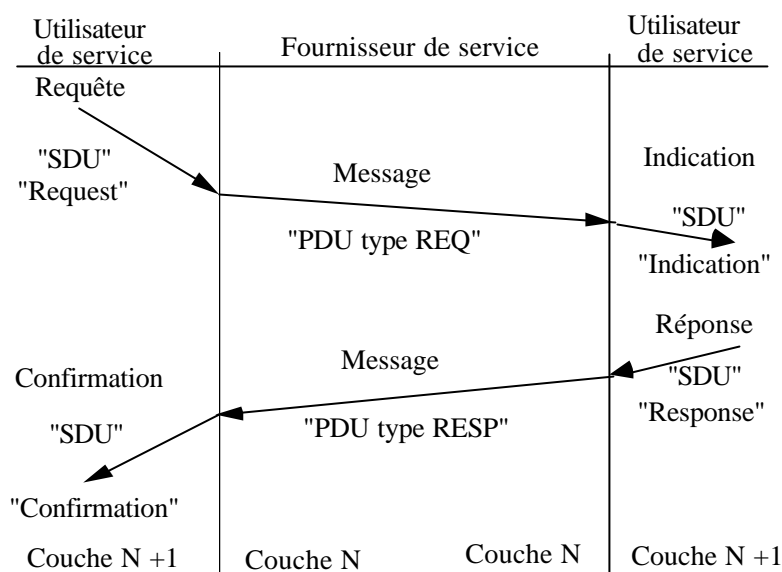
...



# Résumé des différentes notions associées à la structuration OSI



## Exemple de dialogue pour un accord confirmé entre deux entités de niveau N+1



### Primitives

#### Requête ("Request") :

Initialisée par le niveau N+1 pour obtenir un service du niveau N (Exemple : Connect-Request, Data-Request)

#### Indication ("Indication") :

Le niveau N avise le niveau N+1 de l'activation d'un service (Exemple : Connect-Indication, Data-Indication)

#### Réponse ("Response") :

Réponse du niveau N+1 au niveau N sur une indication (Exemple : Connect-Response)

#### Confirmation ("Confirmation") :

Du niveau N au niveau N+1 en terminaison du service requis (Exemple : Connect-Confirmation)

## Structure des informations échangées

### Sur le site demandeur:

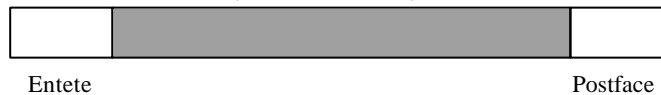
*Niveau N+1* Émet une **Requête** - avec des données (SDU)



*Niveau N* Traite la **Requête**

. Prépare une unité de protocole propre à son niveau en rajoutant des informations (PCI) entête (si nécessaire en queue):

- Entête ("Header") Ex Type, adresse
- Postface ("Trailer") Ex Code détecteur

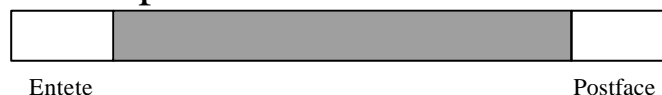


. Transmet l'ensemble au niveau N distant (destinataire).

### Sur le site distant:

*Niveau N*

- Reçoit l'unité de protocole



- Interprète entête et postface (le PCI)
- Émet une primitive **Indication** avec en paramètres les données (SDU)

*Niveau N+1*

- Reçoit l'**Indication** et traite les données



## II.3 Introduction à la spécification des services et des protocoles

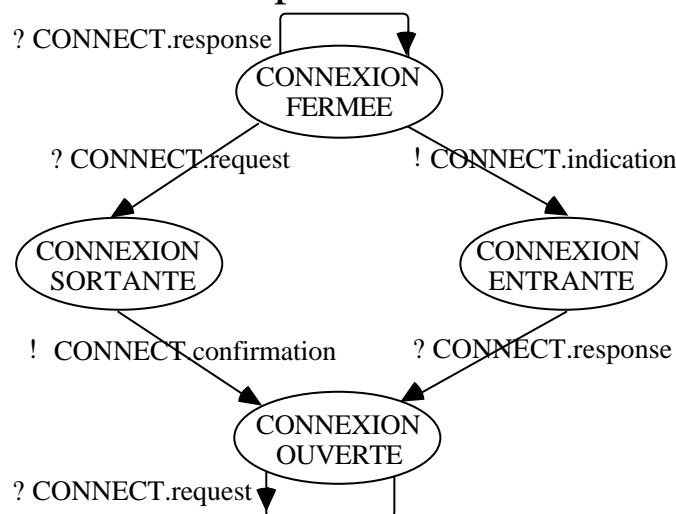
("ASDC Abstract Service Definition Convention")  
("FDT Formal Definition Techniques")

Définition de toutes les séquences correctes d'échanges de d'unités de service et de protocole

### Exemple de solution (la plus répandue)

- Automate d'états finis ("Finite State Machine")
- Notion d'état ("state")
- Notion de transition ("transition") entre états qui correspondent à des émissions d'information !info ou à des réceptions d'information ?info.

Exemple partiel du service pour l'accord confirmé:



=> Besoin de méthodes formelles de spécification et de preuve de comportements corrects

Détection des interblocages, des réceptions non spécifiées.

## **II.4 Le Modèle OSI**

### **II.4.1 Le modèle de référence pour l'interconnexion de systèmes ouverts ("OSI Open System Interconnection")**

#### **A - Un cadre général pour le développement de l'architecture**

##### **Origine**

ISO/IEC 7498 Modèle de référence ("Reference Model")  
Principes architecturaux, différents niveaux.

##### **Développements ultérieurs**

ISO/IEC 7498 - ad 1 Communications en mode non connecté ("Connectionless Mode")

ISO/IEC 7498 - ad 2 Concepts et mécanismes de sécurité ("Security Architecture")

ISO/IEC 7498 - ad 3 Désignation et adressage ("Naming and Addressing")

ISO/IEC 7498 - ad 4 Cadre pour l'administration de réseaux ("Management Framework")

#### **B - Des normes spécifiques qui définissent des réalisations des différents niveaux.**

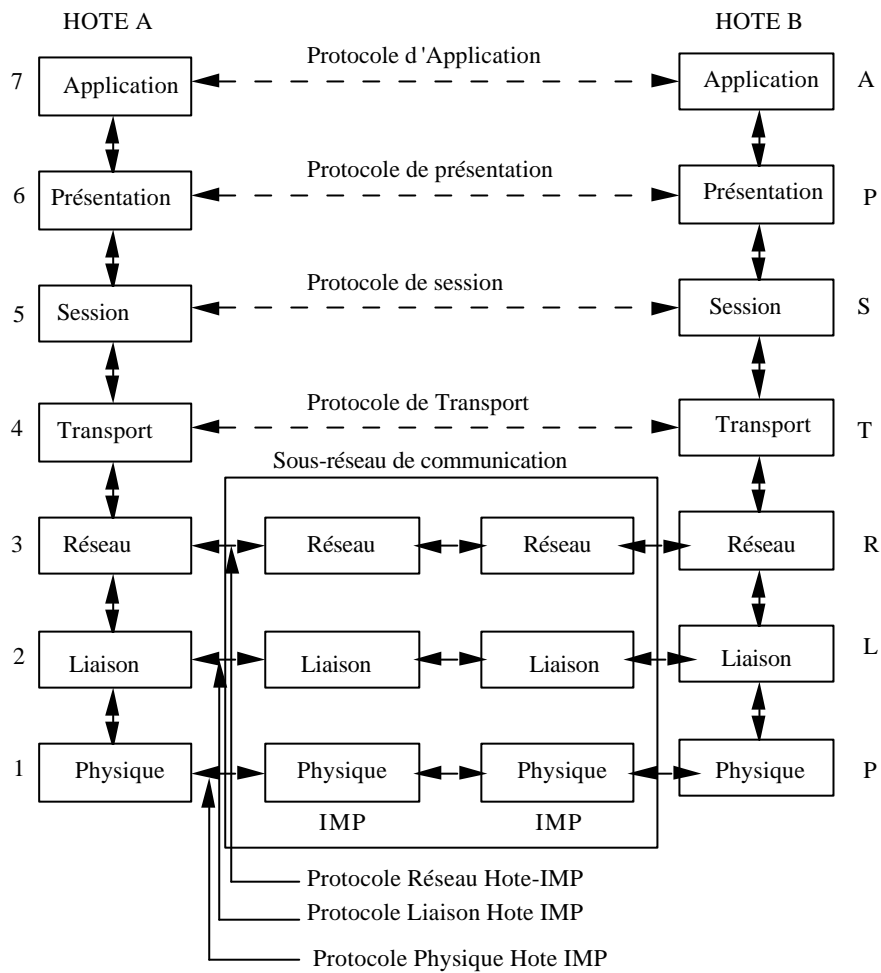
# Le Modèle à sept couches

## Principes de base du modèle

- Les fonctions à exécuter doivent être divisées en **niveaux séparables** du point de vue physique et logique.  
Les fonctions associées dans un niveau doivent avoir une **finalité cohérente**.
- Chaque couche doit contenir **un volume suffisant** de fonctions afin de minimiser le nombre des couches.
- Les protocoles doivent **agir uniquement à l'intérieur** de la même couche.
- Les interfaces entre couches doivent être aussi simples que possible de manière à **minimiser les échanges entre couches**.
- Les couches doivent pouvoir être **modifiées** sans que soient affectés les services qu'elles offrent.
- Une fonction devrait n'apparaître qu'une seule fois.
- L'ensemble doit être efficace en termes de performances



# Schéma du modèle de référence



## II.4.2 Éléments communs rencontrés dans la réalisation des différents niveaux

### Gestion des flots d'informations

#### Ouverture/fermeture de connexions

##### Ouverture ("Connexion establishment")

- . Dans un protocole orienté connexion :  
**phase de délimitation préliminaire d'un échange.**
- **Désignation** du destinataire à atteindre.
- **Négociation** de paramètres de qualité de service.

QOS "Quality Of Service"

Paramètres **qualitatifs**

Exemple: mode de fonctionnement simplex,  
à l'alternat, duplex.

Paramètres **quantitatifs**

Exemple : taux d'erreur résiduel accepté.

##### Fermeture de connexions ("Connexion release")

- . Dans un protocole orienté connexion **phase de délimitation de terminaison d'un échange.**

**Déconnexion abrupte** ("Destructive release")

( avec perte éventuelle des informations en transit).

**Déconnexion ordonnée** ("Orderly release")

(avec transmission des données en cours avant fermeture)

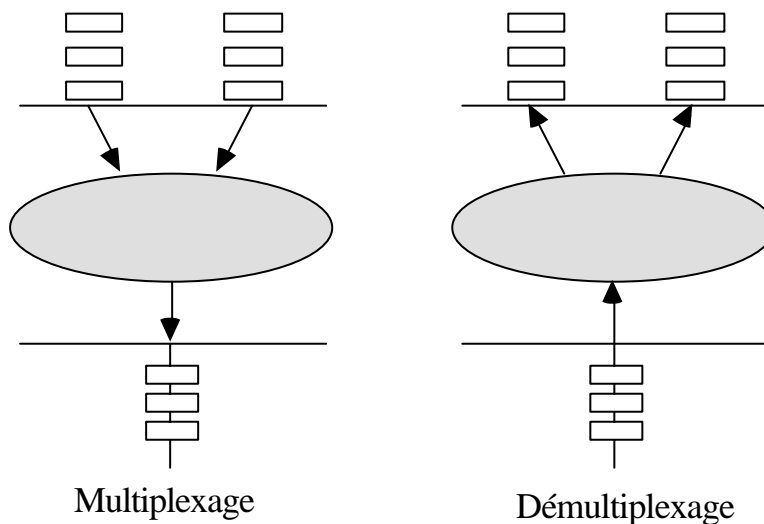
**Déconnexion négociée** ("Graceful" "Negociated release") (avec l'accord des deux parties).

## Multiplexage ("Multiplexing")

**On doit associer à un flot de messages de niveau N+1 un flot de niveau N** (par exemple une connexion de niveau N+1 à une connexion de niveau N).

**Le multiplexage signifie que plusieurs flots de niveau N+1 sont associés à un même flot de niveau N** (opération inverse : démultiplexage).

Problème à résoudre: assembler puis séparer les différents flots de communication au moyen d'identifiants.



### Exemples d'utilisation

Pour des flots de messages appartenant à des communications différentes d'un même usager.

Pour de flots appartenant à des usagers différents.

Pour des flots appartenant à des protocoles différents.

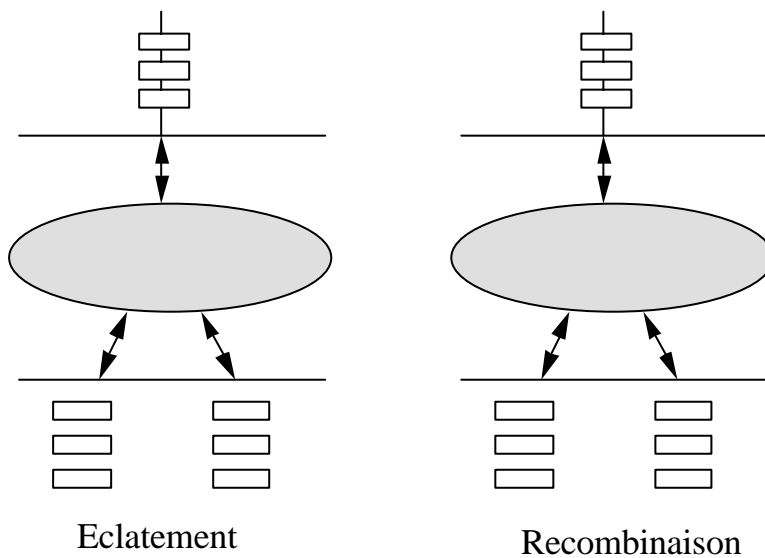
## Éclatement ("Splitting")

Une connexion de niveau N+1 est éclatée sur plusieurs connexions de niveau N.

Exemple: pour améliorer le débit.

Opération inverse : la **recombinaison**.

Problème : **reconstruire la séquence** de la connexion éclatée.



## **Commutation - Routage ("Switching - Routing")**

Acheminer un paquet dans un réseau maillé en visitant des commutateurs successifs.

- déterminer le chemin à suivre (optimalement) pour aller d'un point à un autre.

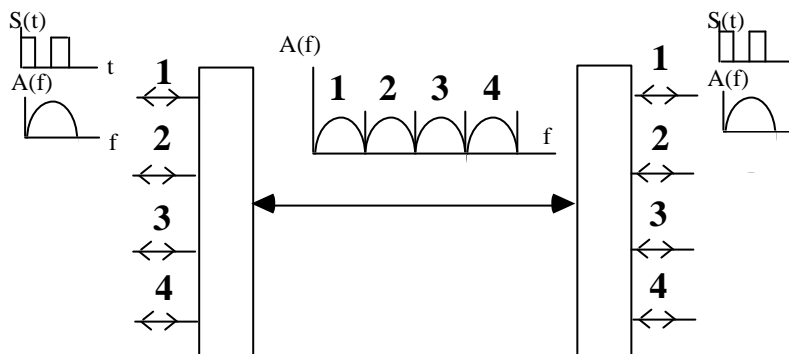
- réaliser (rapidement) les opérations d'aiguillage d'une voie entrante sur une voie sortante.

- gérer des files d'attente associées aux voies en entrée ou en sortie ou au milieu ... pour limiter au maximum les pertes de messages dues à la congestion.

# Les différents types de multiplexage

## Multiplexage à répartition de fréquence

("AMRF Accès Multiple à Répartition de Fréquences")  
("FDMA Frequency Division Multiplexing Access")



### Principes

Le spectre de la voie haute vitesse est découpé en bandes étroites associées à chaque voie basse vitesse.

### Applications

Multiplexage de voies physiques:

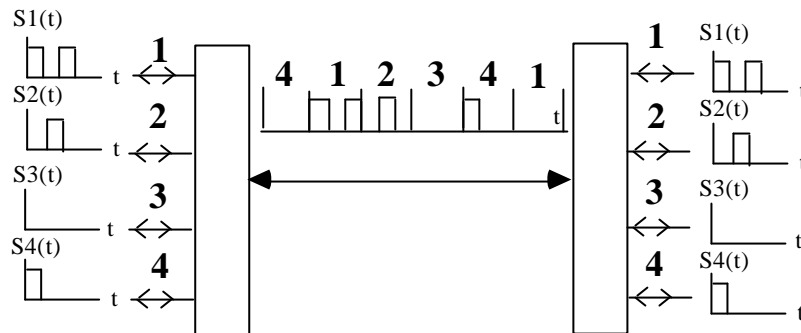
- . réseaux large bande ("Broadband"),
- . réseaux télévision,
- . anciennement en téléphonie,
- . multiplexage en longueur d'onde (Wavelength)

## Multiplexage temporel (à répartition de temps)

### Multiplexage Temporel Synchron

("AMRT Accès Multiple à Répartition de Temps")

("TDMA Time Division Multiplexing Access")



### Principes

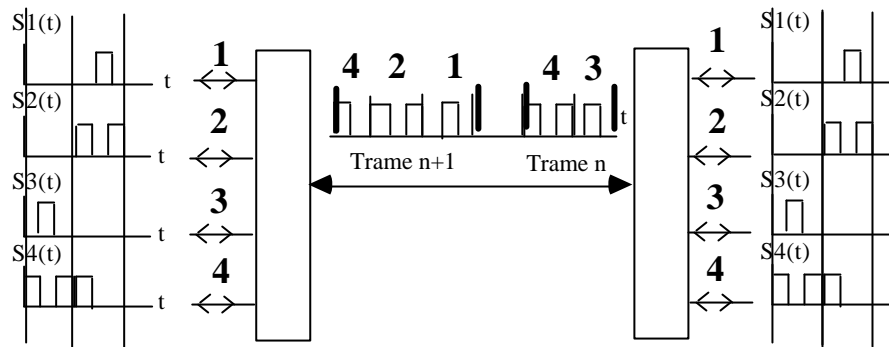
Les intervalles de temps de la voie haute vitesse sont associés successivement à chaque voie basse vitesse.

### Applications

Multiplexage de voies physiques utilisé en téléphonie, RNIS-BE  
Réseau Numérique à Intégration de Services - Bande Étroite

# Multiplexage Temporel Asynchrone Multiplexage Statistique

("ATDM Asynchronous Time Division Multiplexing")



## Principes

Les données sont échantillonnées sur les voies basses vitesses selon leur rythme d'arrivée et sont rassemblées en trames de la voie haute vitesse.

## Applications

Multiplexage de voies utilisé en informatique

- multiplexeurs physiques,
- concentrateurs,
- multiplexage de connexion N+1 sur une connexion de niveau N (réseau sur liaison, ...)

Réseau numérique à intégration de service large bande (RNIS -LB).

"B-ISDN Broadband Integrated Numerical Service"

"ATM Asynchronous Transfer Mode"

"TTA Technique Temporelle Asynchrone"

Multiplexage temporel de cellules.



# Les différents types de commutation

## *La commutation de circuits*

Un chemin permanent (le circuit) est établi entre deux entités communicantes et assure une bande passante fixe.

### **Avantages**

- Ouverture d'un canal totalement disponible et indifférent au type de données transférées.
- Permet de réserver une capacité nette pour des trafics synchrones ou temps réel (voix, image, données à échéances)

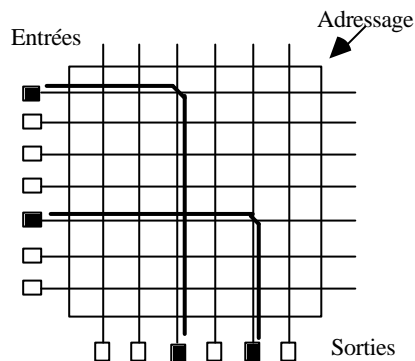
### **Inconvénients**

- La capacité mise à disposition n'est pas toujours adaptée au besoin et peut-être parfois extrêmement mal employée (données informatiques, voix ou image compressées)

# La commutation spatiale (commutation de circuits)

## Principe

Établissement d'un lien métallique permanent au moyen d'aiguillages d'interconnexion N entrées N sorties ( $N^2$  intersections: "crossbar").



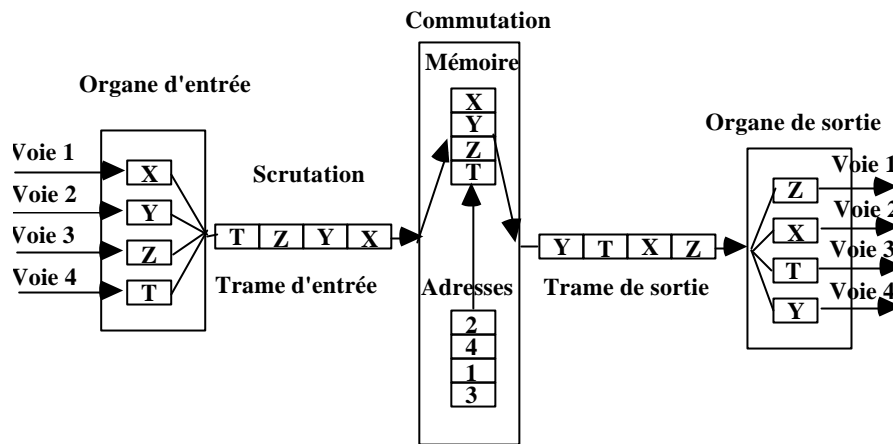
On peut traverser plusieurs commutateurs en cascade pour déterminer un chemin.

Optimisation : les réseaux d'interconnexion

## Utilisation

- Commutation téléphonique ancienne.
- Commutation très haut débit (accès mémoire, commutateurs gigabits, ....)

# La commutation temporelle synchrone (commutation de circuits)



## Principes

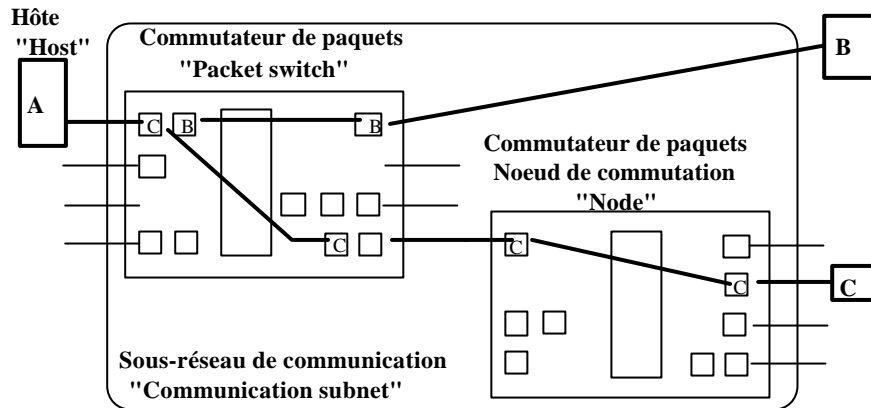
- Chaque entrée est échantillonnée de façon synchrone (n fois par secondes) pour constituer un multiplex (trame d'entrée)
- Lorsqu'une trame est présente (n fois par secondes) elle est écrite en mémoire.
- Une table de correspondance entre positions d'entrée et position de sorties permet de réordonner la trame
- L'affichage en sortie s'opère après démultiplexage selon les nouvelles positions et assure la fonction de commutation.

## Utilisations

- Téléphonie numérique, RNIS.

# La commutation de paquets (Commutation temporelle asynchrone)

## Les circuits virtuels ou les datagrammes



## Principes

- Les paquets sont des éléments d'informations qui comportent une entête avec l'adresse du destinataire.
  - Les paquets circulent sur des voies physiques en multiplexage asynchrone et arrivent de façon asynchrone dans des calculateurs de commutation.
  - Ils sont - **mis en file d'attente** en mémoire après acquisition sur une voie d'entrée,
    - **aiguillés vers une file de sortie** en fonction du destinataire à atteindre.
    - **renvoyés** vers un adjacent ou le destinataire.
- Technique de stockage et retransmission ("**store and forward**")

# Commutation de paquets

## Avantages

- Apporte une grande adaptabilité aux débits soumis par les usagers.
- Permet d'optimiser les voies de communication.

## Inconvénients

- L'opération de commutation est plus complexe qu'en commutation de circuits et les débits commutés sont plus faibles.
- Les trafics voix image ont des caractéristiques particulières qui rend délicate l'utilisation de la commutation temporelle asynchrone.

## Définitions

**Commutation de messages** : ancienne approche de commutation d'éléments de taille quelconque => problèmes d'allocation de tampons

**Commutation de paquets** : la taille des paquets est bornée et permet une bonne gestion mémoire.

**Commutation de cellules** : la taille des cellules est strictement fixée => efficacité maximale.

## Applications

- Commutation informatique (X25, IP, IPX...).
- Réseau numérique à intégration de service large bande (RNIS -LB).
  - "ATM Asynchronous Transfer Mode"
  - Commutation temporelle de cellules

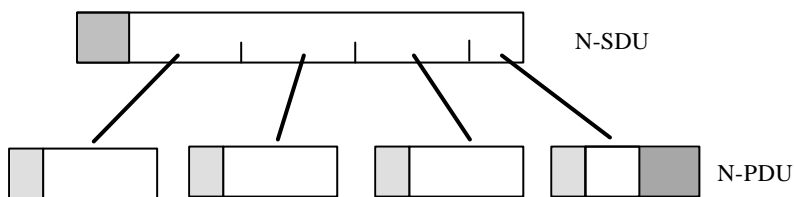
# Transfert des données

## Segmentation ("Fragmentation")

Dans le cas où une information usager à transporter (N-SDU) est trop grosse pour la taille (d'un maximum imposé) des messages du niveau (N-PDU)

=> Obligation de la découper en morceaux.

Opération inverse réassemblage

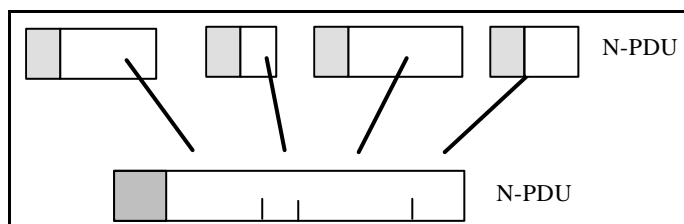


## Groupage ("Concaténation")

Dans le cas où une information à transporter (N-PDU) est trop petite par rapport à la taille des messages qui optimisent les performances du niveau (taille fixe ou d'un minimum donné)

=> Optimisation possible en regroupant des morceaux N-PDU.

Opération inverse dégroupage



## **Techniques de compression ("compression")**

Réduire le volume de données échangées lorsque celles-ci présentent des redondances.

## **Transfert de données normales ("Normal data flow")**

Le type de données habituellement échangées.

## **Transfert de données expresses ("Expedited data transfer")**

Un autre type de données devant circuler rapidement (exemple alarmes, exceptions).

## **Contrôle de flux ("Flow control")**

- En raison de l'hétérogénéité des vitesses de traitement, de la capacité des tampons => Adaptation de la vitesse de l'émetteur à celle du récepteur.

### ***Contrôle de flux inter sites ("Peer flow control")***

Entre deux niveaux appariés (défini dans le protocole)

### ***Contrôle de flux inter niveaux ("Inter layer ...")***

Entre un niveau N+1 et le niveau N  
dépendant de l'implantation fournisseur.

## **Contrôle d'erreur ("Error control")**

Maintenir l'intégrité des données transférées.

En présence de bruit: sur les voies, dans les mémoires, ...

En cas de perte due à l'incapacité de stocker

- détecter la modification ou la perte des données échangées (codes détecteurs d'erreurs)
- retransmettre en cas d'erreur.

(Problème d'intégrité en termes de sécurité

Empêcher un intrus de modifier les données).

## **Contrôle de séquence ("Sequencing")**

Une couche N doit en général préserver l'ordre des SDU qui lui sont confiés par le niveau N+1.

Si le protocole employé ou la couche inférieure peuvent dupliquer ou déséquencer les informations:  
il faut restituer l'ordre des soumissions.

## **Contrôle de congestion ("Congestion control")**

Éviter en présence d'une surcharge de trafic l'écroulement (la destruction de paquets dans le réseau conduisant à une décroissance du trafic utilisateur transporté).



## **Autres fonctions**

### **Désignation ("naming")**

Définition de la structure des noms et adresses réseaux, de leur attribution et de la façon d'établir une liaison entre un nom et une adresse.

### **Administration réseau ("network management")**

Assurer les tâches de suivi des configurations.

- Gestion des configurations
- Gestion des pannes
- Gestion de la facturation
- Gestion des performances
- Gestion de la sécurité (voir plus loin)

### **Sécurité ("security")**

Assurer au moyen de cryptages la résistance des communications face à des actions de malveillance:

- Intégrité.
- Authentification
- Confidentialité

**II.5**  
**PRÉSENTATION DES FONCTIONS DES**  
**DIFFÉRENTS NIVEAUX**

## NIVEAU PHYSIQUE

. Il fournit les moyens nécessaire à l'activation au maintien et à la désactivation des connexions physiques destinées à la transmission de suites binaires.

### - Mécaniques

Exemples: connecteurs, forme des prises, utilisation des broches pour les différents signaux.

### - Électriques

Exemples: modulations, utilisation des niveaux disponibles pour coder un bit, durées.

### - Procéduraux

Exemples: protocoles de synchronisation entre l'émetteur et le récepteur, multiplexage, transmission bidirectionnelle, à l'alternat.

**Notion de niveau physique** qui comporte la globalité de la chaîne de transmission,

Exemples :niveau physique ethernet sur paire torsadée (10 Base T), Liaison spécialisée Transfix

**Notion d'interface standard** entre la voie physique de transmission et le système informatique (interface ETTD-ETCD). Celle-ci en constitue un élément.

Exemples : EIA-RS 232 C, CCITT V24 ou X21, interface ethernet (interface MAU).

## NIVEAU LIAISON

. Il assure le transfert d'informations entre deux calculateurs reliés par une voie physique.

. Selon le support physique et les options de conception une partie des fonctions suivantes est offerte.

### Voies multipoints

Problème de partage de l'accès au médium  
("MAC Medium Access Control")

### Exemples de stratégies de partage

#### . **Compétition**

Réseau ethernet ISO 8802-3 (10 Mb/s et 100 Mb/s)

#### . **Gestion de boucles à jeton**

Boucle à jeton IBM ISO 8802-5, FDDI ANSI X3T9

#### . **Réservation statique**

Multiplexage temporel synchrone

#### . **Réservation dynamique**

DQDB "Distributed Queue Dual Bus"

#### **Scrutation**

Exemples : réseaux de terrain industriels (FIP)

## Communications point à point

Gestion de liaison entre deux voisins reliés par une voie de type quelconque

. **Délimitation/mise en correspondance d'unités de données.**

. **Multiplexage** (de flots de données d'origines différentes).

. **Contrôle d'erreur** (transformer une voie bruitée en une voie de taux d'erreur acceptable).

. **Contrôle de flux.**

. **Contrôle de séquence.**

. **Établissement et libération de connexions.**

. **Fonctions annexes d'administration** de liaison

- *L'identification des entités de liaisons*

- *La gestion de paramètres de configuration*

- *La surveillance de la voie.*

### **Exemples d'implantations**

Niveaux liaisons en connexion type HDLC

("High Level Data Link Communication")

LAPB, "Link Access Protocol B"

LLC2, "Logical Link Communication 2"

LAPD, "Link Access Protocol D"

Niveaux liaison sans connexion

PPP, Internet "Point to Point Protocol"

LLC/SNAP, "Logical Link Communication 1

/Sub Network Access Protocol".

## NIVEAU RÉSEAU

**Contrôle le fonctionnement du sous-réseau de communication principalement en déterminant comment les paquets sont routés d'un ordinateur ("hôte") source vers un ordinateur ("hôte") destinataire.**

L'un des niveaux les plus complexes (avec application).  
Selon les options de conception une partie des fonctions suivantes est réalisée

- **Routage (en point à point ou en diffusion).**

Échange d'unités de données entre sites reliés par un réseau (indépendamment de la technologie du réseau, local, maillé,...). Suppose un mécanisme d'adressage uniforme au niveau du réseau.

- **Contrôle de congestion**

- **Segmentation.**

- **Multiplexage.**

(des connexions de réseau sur des connexions de liaison).

- **Contrôle de séquence.**

- **Gestion des connexions.**

- **Contrôle d'erreur.**

(détection et correction des erreurs d'hôte à hôte).

- **Contrôle de flux.**

## **Exemples de services et de protocoles de réseau**

### **Internet**

Protocole **IP-V4/V6** ("**Internet Protocol version 4 et 6**")

### **IBM SNA**

Niveau "**Path Control**"

#### **Réseaux publics de transmission de données:**

Mode connecté

"CONP" "Connection Oriented Network Protocol"  
ISO/IEC 8348 "Network Service Definition"

"CONS" "Connection Oriented Network Service"  
ISO/IEC 8208 **X25 Niveau 3 "PLP Packet Layer Protocol"**

Mode non connecté

"CLNP" "ConnectionLess Network Protocol"  
"CLNS" "ConnectionLess Network Service"

### **Réseaux Novell**

"IPX" Internetwork Packet eXchange protocol, basé  
sur le protocoleXNS (Xerox Network System)

## NIVEAU TRANSPORT

Le premier des niveaux utilisable directement par l'utilisateur final pour développer des applications.

- . **Assure un service de transmission fiable entre processus (de bout en bout, "end to end").**
- . **Il résume les fonctions de télécommunications.**

### Selon les options de conception

- **Gestion des connexions.**

Protocoles en mode connecté avec ou sans négociation de qualité de service ou non connecté .

- **Multiplexage/éclatement** (des connexions de transport sur des connexions de réseaux).

- **Contrôle d'erreur.**

- **Contrôle de flux.**

- **Contrôle de séquence.**

- **Segmentation.**

### Exemples de protocoles de transports:

#### Internet

TCP "Transmission Control Protocol".

UDP "User Datagram Protocol".

#### Novell

SPX "Sequenced Packet eXchange"

En fait XEROX XNS: SP Protocol

#### IBM SNA

Niveau "Transmission Control"

#### Transport OSI

"Connection oriented transport service" ISO/IEC 8072

"Connection oriented transport protocol" ISO/IEC 8073



## NIVEAU SESSION

. Dans sa définition de base OSI le niveau session structure et synchronise le dialogue entre deux entités communicantes.

### L'approche OSI.

. Le transport offre **une voie logique** de communication ("un tube") sans organisation spécifique des données échangées.

. Le mode de communication utilisé est **le mode message asynchrone**.

. **La session** permet de structurer les échanges pour y ajouter de la **tolérance aux pannes et de la synchronisation**

- **Activités** (travail important)
- **Dialogue** (partie d'un travail)
- Notion de **point de synchronisation** pour délimiter des parties d'un échange en vue de la reprise.

### Difficulté majeure de l'approche OSI

La session OSI a été définie **prématurément** alors que la recherche dans le domaine de la structuration et de la synchronisation répartie était peu avancée.

**D'autres fonctions de synchronisation** sont apparues et ont été placées au **niveau application**.

### Référence des normes

CCITT X215 ISO 8026 Spécification du service de session

CCITT X225 ISO 8027 Spécification du protocole de session

## L'approche APD Appel de Procédure Distante "RPC Remote Procedure Call"

. Le mode de communication du transport étant **le mode message asynchrone**, la session est définie pour offrir à l'utilisateur un mode de dialogue de type synchrone.

=> Permettre à un usager d'exécuter une procédure ou une fonction sur un autre site

. en lui passant dans un message d'appel des paramètres d'appel

. et en recevant en retour des paramètres résultats.

. Problème délicat

Assurer en environnement réparti une sémantique pour l'appel de procédure distante voisine de celle connue en univers centralisé.

Exemples:

SUN-OS : SUN Operating System

Protocole RPC "Remote Procedure Call"

OSF-DCE : "Open Software Foundation"

"Distributed Computing Environment"

OMG-CORBA: "Object Management Group"

"Common Object Request Broker Architecture"

## NIVEAU PRÉSENTATION

. Le niveau présentation est défini pour gérer la représentation (le codage) des données échangées de telle façon que même si différents sites utilisent des représentations différentes ils peuvent utiliser les données transmises.

### Les conversions

- . Nécessaire pour tous les types de données
  - Types chaînes de caractères (les premiers types de conversion).
  - Types numériques (entiers, flottants, ...)
  - Types complexes (construits).

#### Définition de deux notions.

**Syntaxe abstraite** permettant la définition d'une grande variété de structures de données (analogue de la syntaxe de définition de types dans un langage évolué).

**Syntaxe de transfert** : Une représentation unique dans le réseau utilisée pour transférer les données.

Exemples de niveau présentation:

- Réseaux publics de transmission de données:

Syntaxe abstraite ASN1 : CCITT **X208**

Syntaxe de transfert: CCITT **X209**

Protocole de présentation avec connexion **ISO 8823**

Protocole de présentation sans connexion **ISO 9576**

- **SUN-OS**

Protocole **XDR** : "eXternal Data Representation".

- **IDL** "Interface Definition Languages" DCE, CORBA

## NIVEAU APPLICATION

. Le niveau application est défini pour fournir à l'utilisateur des fonctions dont il a besoin couramment.

- **en termes d'un cadre de développement** d'une application informatique répartie (structuration objet),

- **en termes de "bibliothèques" de protocoles** (fonctions réseaux) prédéfinies qui déchargent l'utilisateur de travaux répétitifs de programmation d'applications souvent utilisées.

## Protocoles d'usage général : approche OSI

Problèmes génériques

. Gestion des connexions ou associations,

. Transfert fiable,

Problèmes de synchronisation non résolus

au niveau session

. Appel de procédure distante,

. Validation à deux phase.

## Protocoles spécifiques

### Le transfert de fichiers

**Objectifs** : Déplacer des fichiers (plats en général d'un site à un autre). Très nombreux protocoles proposés

**Exemples** : Internet FTP "File Transfer Protocol"  
OSI/IEC FTAM "File Transfer Access and Management"

### L'accès aux fichiers distants

**Objectifs** : Accès unifié en univers réparti à différents fichiers (réalisation des requêtes d'accès).

**Exemples** : SUN-OS NFS "Network File System"  
OSI/IEC FTAM "File Transfer Access and Management"

### La gestion transactionnelle

**Objectifs**: Cohérence et la persistance de données en univers réparti => assurer le maintien cohérent d'un ensemble de données réparties en présence de pannes.

**Exemples** : En univers Ouvert  
OSI/IEC TP "Transaction Processing"  
En approche propriétaire: moniteurs transactionnels  
IBM CICS

## L'accès aux bases de données distantes

**Objectifs** : Permettre à un client d'accéder à une base de données distante (le plus souvent au moyen de requêtes SQL)

**Exemples** : En univers Ouvert  
OSI/IEC RDA "Remote data Access"

Normalisation de facto

ODBC : Microsoft "Open Data Base Connectivity"

Proposition du groupe SAG "SQL Access Group" d'un ensemble de requêtes CLI "Call level Interface".

IDAPI : "Integrated Database Application Programming Interface" Autre proposition d'un consortium Borland, IBM, Novell, Wordperfect.

Produits propriétaires

Exemple EDA-SQL de Information Builders Inc

Interface de communication API/SQL connectant de très nombreuses bases de données sur des plates formes clientes.

## La désignation

**Objectifs** : Gérer des annuaires permettant à un utilisateur de retrouver des caractéristiques (principalement l'adresse réseau) d'un correspondant.

**Exemples** : Internet DNS "Domain Name System"  
OSI/IEC DS "Directory Services"

## La messagerie

**Objectifs** : Permettre d'échanger du courrier électronique entre usagers

**Exemples** : Internet SMTP "Simple Mail Transfer Protocol"

OSI/IEC MHS "Message Handling Systems"

## L'échange de données informatisées

**Objectifs** : Permettre d'échanger des documents administratifs standards sur des réseaux de transmission de données (commandes, factures, ...) entre agents économiques.

**Exemples** : Normes EDI "Electronic Data Interchange"

## L'échange de documents électroniques

**Objectifs** : Permettre d'échanger des documents généraux structurés de types particuliers (lettre, journaux, livres, ...).

**Exemples** : ODA "Office Document Architecture" et très nombreuses autres propositions.

HTML « Hyper Text Markup Language »

## Autres fonctions

Le transfert de travaux

Le terminal virtuel

# **CONCLUSION :**

## **Le modèle de référence OSI**

- Une situation délicate avec de très nombreux types d'attaques:

### **Les critiques directes du modèle**

- Les niveaux ne sont pas également remplis (bien utilisés).
- Certaines fonctions ne sont pas placées au bon endroit ou sont déplacées au cours du temps.

=> Évolution historique

- Certaines fonctions peuvent être répétées inutilement à plusieurs niveaux (selon les choix des profils) ?

Contrôle d'erreur ?

Contrôle de flux ?

=> Pas forcément inutile.

- Le modèle OSI est dominé par une approche télécommunications et n'intègre pas assez les approches informatiques de structuration des applications.

Exemple : Problème de la gestion événementielle des applications en mode message => Il existe des solutions.

=> Évolution vers l'approche objet

### **Les approches non prises en compte par l'OSI (en cours)**

Communications sur réseaux haut débits

Les approches systèmes répartis et micro-noyaux

...



## II.6 Interconnexion de réseaux

- Assurer des communications entre calculateurs (ou ES "End Systems") connectés à des réseaux différents.
- Chaque réseau est en fait un sous-réseau d'un ensemble (d'un réseau de sous-réseaux).
- Les sous-réseaux sont interconnectés par des routeurs ("Routers" ou IS "Intermediate Systems").

### Problèmes particuliers

- . Existence de **deux niveaux de routage**
  - Routage **propre** à chaque sous-réseaux
  - Routage **inter sous-réseaux**:  
détermination du prochain routeur (IS)  
à traverser pour atteindre le destinataire final.
- . Routage inter sous réseaux :  
possibilité d'emprunter des chemins passant par  
des sous-réseaux différents (optimisation difficile)
- . **Problèmes d'adressage**
  - Adressage universel**  
commun aux divers sous- réseaux.  
Conversions d'adresses
- . **Conversion des formats** utilisés dans les réseaux.

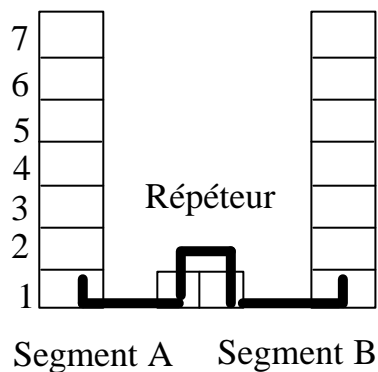
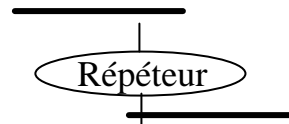
### Exemple type: IP Internet Protocol

## II.6.1 Interconnexion au niveau physique Répéteurs ("Repeaters")

On prolonge un médium de communication dont la longueur est insuffisante.

Essentiellement pour une voie physique multipoint (réseau local).

Le signal  
est répété  
d'un segment  
sur l'autre.

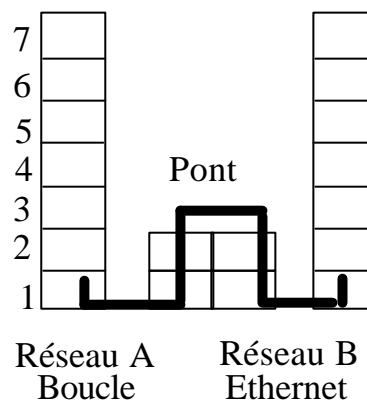
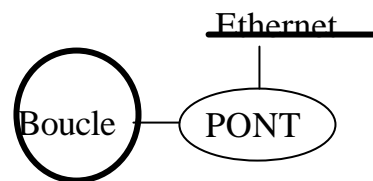


## II.6.2 Interconnexion au niveau liaison Ponts ("Bridges")

On capture dans un système informatique (assez complexe) toutes les trames de niveau liaison d'un réseau pour les transférer à un autre réseau.

Pour deux réseaux locaux dont les systèmes d'adressage sont compatibles et qui sont sans homonymie.

Les trames  
sont  
récopiées  
d'un réseau  
sur l'autre.



**Exemple: Interconnexion au moyen d'un pont**

# **Extensions de la notion de pont**

## **Notion de pont filtrant**

Raccorde deux réseaux en filtrant les adresses.

- Tout ce qui doit passer d'un réseau sur l'autre traverse le pont),
- Toute communication propre à un réseau le reste.

## **Notion de commutateur de réseaux locaux**

Raccorde plusieurs réseaux en filtrant les adresses et en assurant les fonctions de commutation d'un tronçon vers un autre.

- Tout ce qui doit passer d'un réseau sur l'autre traverse le commutateur),
- Toute communication propre à un réseau le reste.

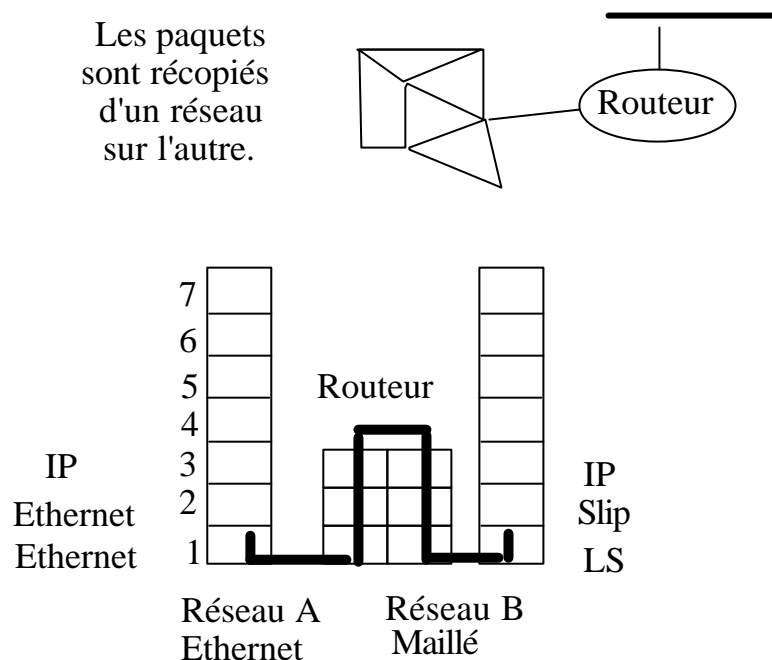
## II.6.3 Interconnexion au niveau réseau

### Routeurs ("Routers")

### Passerelles ("Gateways")

On route des paquets (de niveau réseau) à destination d'un autre réseau.

Pour des réseaux disposant d'un système d'adressage d'interconnexion.



### Routeur multiprotocoles

Raccorde des réseaux de standards différents

- IP.
- X25
- Apple Talk, ...

## **II.6.4 Interconnexion aux niveaux supérieurs Convertisseurs de protocoles ("Protocol converters")**

Adaptation des messages d'un format et selon un protocole donné à un autre format ou un autre protocole voisin.

Exemple: transformation de dialogues minitel/videotex (affichage et saisie) en format du WEB.

**III**

**EXEMPLES D'ARCHITECTURES**

**DE RÉSEAUX**

## **III.1**

# **RÉSEAUX DE TRANSMISSION**



### **III.1.1 Réseaux dorsaux ("Backbone Networks")**

**Définir l'infrastructure de télécommunications:**

- permettant d'acheminer les différents trafics qu'un opérateur doit supporter.
- voies téléphoniques mais également tous les différents services de transmission intégrés
- sur tous les types de supports physiques.

### **Hiérarchie numérique plésiochrone (PDH "Plesiochronous Digital Hierarchy")**

**Essentiellement multiplexage temporel de voies téléphoniques MIC (64kb/s).**

- . Synchronisation en présence de délais de propagations et décalages d'horloge.
  - . A tous les niveaux les cycles des trames sont indépendants (caractère plésiochrone)
    - Verrouillage de trame et technique de bourrage.
    - Norme G702 Hiérarchie des débits
    - Norme G703 Principes de codage
- => Problème pour extraire ou insérer des circuits de faibles débits dans des trames de débits élevés.

<b>Structure</b>	<b>Débit numérique</b>	<b>Capacité en circuits à 64 kb/s</b>	<b>Avis CCITT</b>
Niveau-1	2 048 Kb/s	30	G.704
Niveau-2	8 448 Kb/s	120	G.742
Niveau-3	34 368 Kb/s	480	G.751
Niveau -4	139 264 Kb/s	1920	G.751

# Technologie numérique synchrone (SDH "Synchronous Digital Hierarchy" G707)

## Objectif

Résoudre les problèmes de la hiérarchie plésiochrone pour permettre de définir un réseau d'infrastructure souple.

Travaux dérivés de **SONET** ("Synchronous Optical Network") (Bellcore).

- **Multiplexage temporel**
- **Brassage de voies (souple)**

Utilisation de **pointeurs** acheminés dans les trames qui permettent de compenser **les problèmes de variation de phase du signal** (délais de propagation et décalages d'horloges).

L'accès à des circuits de faibles débits dans des trames de débits élevés se fait simplement permettant les modifications rapides de configuration.

=> Intégration de réseaux commutés, liaisons spécialisées  
, ...

STM-1	:	155,520 Mb/s
STM-4	:	622,080 Mb/s
STM-16	:	2 488,320 Mb/s

## **III.1.2 Réseau Téléphonique**

**(RTC Réseau Téléphonique Commuté)**

### **Objectif**

Principalement transport de la voix parlée mais également ouverture au transport de données numériques (informatique, fax, images, ...).

#### **Interface de l'utilisateur**

- Canaux de 300 à 3400 Hertz de bande passante.
- Protocole de signalisation dans la bande
  - . Émission d'appel (décrochage de combiné)
  - . Présélection: reconnaissance d'appel par le standard.
  - . Enregistrement du numéro demandé
  - . Définition du routage, acheminement de la demande
  - . Surveillance libération et taxation de l'appel.

#### **Fonctionnement interne**

- Multiplexage de voies MIC 64 Kb/s
- Réseau de signalisation
  - Systeme de signalisation n° 7
- Très grand nombre de problèmes à résoudre
  - Tolérance aux pannes
  - Équilibrage de charge
  - Acheminement international.

## **III.1.3 Réseaux publics à commutation de paquets de transmission de données numériques (X25)**

### **Objectif**

Fournir un service de transmission de données informatiques à commutation de paquets.

### **Hiérarchie des protocoles**

Le niveau physique, liaison , réseau ("Transpac").

<b>X25 PLP "Packet Layer protocol"</b>
<b>X25 LAPB "Linkage Access Protocol B"</b>
<b>X21</b>

## III.1.4 Réseau Numérique à Intégration de Services (RNIS) (ISDN "Integrated Service Digital Network")

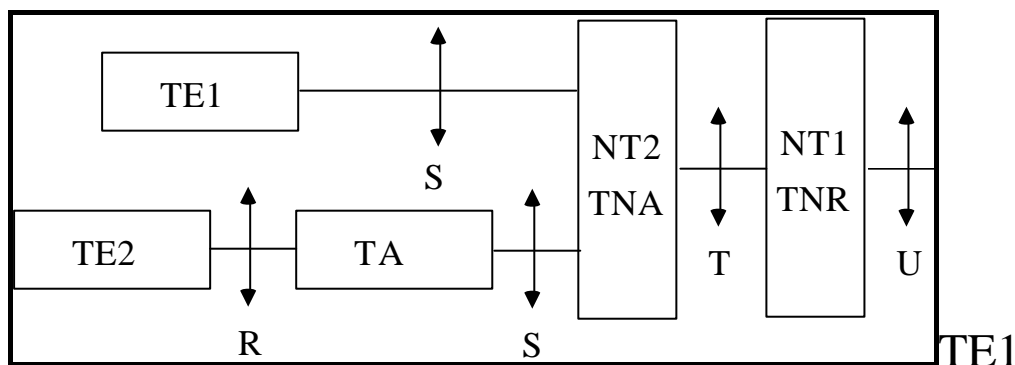
### Objectif

Évolution du réseau téléphonique vers le tout numérique (même l'interface usager).

Offrir un accès multimédia (voix, images, données) dans le domaine de la bande étroite 64kb/s à 2 Mb/s.

Améliorer la gestion de la signalisation

### Organisation des équipements



("Terminal Equipment 1")

Terminal numérique aux normes RNIS

TE2 ("Terminal Equipment 2")

Terminal aux normes anciennes

TA ("Terminal Adaptor") Adaptateur

NT2 ("Network Termination 2") Terminaison Numérique d'abonné . Autocommutateur privé ou régie d'abonné.

NT1 ("Network Termination 1") Terminaison Numérique de réseau . Attachement à la ligne de l'abonné.

# Hiérarchie des protocoles RNIS

Le RNIS est utilisable selon deux types de canaux:

## Transmission sur canal B (à 64 Kb/s)

CCBT Circuit commuté sur canal B transparent  
(transmission de données numériques)

CCBNT Circuit commuté sur canal B non transparent  
(transmission des signaux 300 à 3400 Hz)

## Transmission sur canal D (à 16 Kb/s)

- . soit en mode paquet X25
- . soit pour le protocole D de signalisation.

<b>X25 PLP</b>	<b>Q930-I451 Protocole D</b>
<b>Q921-I441 LAPD</b>	
<b>I431 "Interfaces S/T"</b>	

# III.1.5 Réseau ATM

## TTA Technique Temporelle Asynchrone

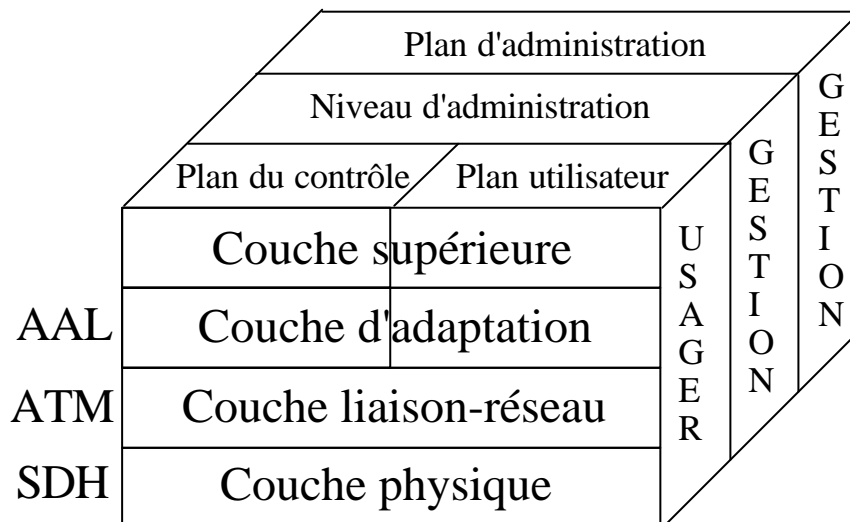
### "ATM Asynchronous Transfer Mode"

## Objectif

Même ambition que pour le RNIS: **l'intégration de services**

=> **RNIS-LB large bande B-ISDN**  
**"Broadband Integrated Service Data Network"**

## Hiérarchie des protocoles



## **III.1.6 Réseaux Locaux** **("Local Area Networks")**

Définir des moyens de communication d'entreprise à **débit élevé** (de 10 à 100 méga bits/s) **sur des distances de quelques kilomètres.**

Le niveau liaison dans les réseaux locaux de type IEEE802 ou ISO 8802 comporte essentiellement la résolution du problème d'accès au médium ("MAC Medium Access Control")

LIAISON	Point à point	8802 - 2
	Accès au médium	8802 - 3 à N
PHYSIQUE	8802 - 3 à N	

### **Exemples de quatre architecture de réseaux locaux**

#### **8802-3 Ethernet (10 Mb/s)**

Réseau à compétition sur bus

#### **8802-4 Bus à jeton ("Token Bus")**

Réseau à coopération sur bus

#### **8802-5 Boucle à jeton ("Token Ring") (4, 16 Mb/s)**

Réseau à coopération en boucle

#### **ANSI X3T9 FDDI :**

**("Fiber Distributed Data Interface") (100 Mb/s)** Réseau à coopération en boucle



## **III. 2 ARCHITECTURES DE RÉSEAUX GÉNÉRAUX**

## **III.2.1 Architecture INTERNET**

- Le plus grand réseau au monde de très loin  
Promis à un avenir très important.
- Héritier du réseau ARPANET  
Représente une expérience considérable en matière de réseaux.
- Architecture de réseau antérieure au modèle OSI (non conforme), il en a quand même différents aspects
  - => Architecture à sept couches similaires OSI.
  - => Le protocole IP a été normalisé OSI.
- Existence de très nombreux produits "contribués" (gratuits) certains expérimentaux, d'autres d'excellent niveau industriel.

### **ARPANET**

Construction du réseau ARPANET à partir de 1969 sous forme de contrats de entre l'ARPA (ministère de la défense des USA) et des organismes (universitaires).

Développement de protocoles de transmission (couches basses) aujourd'hui abandonnés.

Puis développement d'applications : beaucoup sont encore utilisées( transfert de fichiers FTP, accès distant TELNET,... ).

### **INTERNET**

Développement de couches réseaux nouvelles (vers 1980) dans le monde UNIX pour l'utilisation d'ARPANET et des réseaux locaux.

Version UNIX Berkeley BSD-4.2 avec TCP-IP.

### **SUN-OS**

Trois protocoles formant un ensemble couches hautes cohérent.

# Organisation de la pile INTERNET

	Applications TCP/IP directes		Applications pile SUN/OS
<b>7. Application</b>	EXEMPLES		<b>NFS: "Network File System"</b>
<b>6. Présentation</b>	<b>SMTP "Simple Mail Transfer Protocol"</b>	<b>FTP: "File Transfer Protocol"</b>	<b>XDR: "External Data Representation"</b>
<b>5. Session</b>			<b>RPC: "Remote Procedure Call"</b>
<b>4. Transport</b>	<b>TCP: Transmission Control Protocol (connecté)</b> <b>UDP: User Datagram Protocol (non connecté)</b>		
<b>3. Réseau</b>	<b>IP: Internet Protocol</b>		
<b>2. Liaison</b>	<b>Encapsulation IP (sur LAN ou liaisons SLIP,PPP)</b>		
<b>1. Physique</b>	<b>Pratiquement tout support de transmission</b>		
	<b>Réseaux Publics</b>	<b>Lignes spécialisées Point à Point</b>	<b>Réseaux Locaux</b> <b>Réseau téléphonique RNIS, ATM</b>

## Liste de services offerts par l'Internet au niveau application

- **Courrier électronique**  
("Electronic Mail") **mail**
- Soumission de **travaux**  
("Remote Login") **telnet, rlogin , rsh**
- **Informations** sur les usagers  
("Finger Services") **finger**
- **Forum** d'utilisateurs  
("User's network" "Usenet" "News") **news**
- **Accès** anonymes libre service/transfert de fichiers  
("File Transfer Protocol", Anonymous ftp) **ftp**
- **Dialogues** interactifs  
("Talk facilities") **talk**
- Accès à des informations à base de **menus**  
("Menu based information") **gopher**
- Accès à des bases de **données indexées**  
("Search Indexed Databases") **wais**
- Accès à des informations **hypertextes**  
("World Wide Web") **web.**

## **Conclusion**

### **Introduction aux réseaux**

- Domaines des réseaux informatiques : un axe essentiel du développement des techniques numériques : données, voix, images.
- Domaine vaste et complexe qui comporte aussi bien des techniques de télécommunications que des aspects de plus en plus sémantiques des interactions entre machines.
- Grande hétérogénéité des propositions malgré des efforts de normalisation et d'homogénéisation multiples.
- Evolutivité des concepts et des techniques très importante.
- Importance considérable des perspectives de développement industriel.

**Niveau**

**Physique**

## **I. Transmission sur un canal**

1. Transmission en bande limitée
2. Transmission en présence de bruit
3. Détection et correction des erreurs
4. Représentation des signaux

## **II. Éléments de technologie**

1. Les contrôleurs
2. Les interfaces standards
3. Les modems
4. Les voies de communication
5. Les réseaux au niveau physique
  - Le réseau téléphonique commuté
  - Les réseaux PDH et SDH
  - Le réseau RNIS

# INTRODUCTION

## **Objectifs du niveau physique**

Transmission effective des informations binaires sur une voie physique en s'adaptant aux contraintes du support physique utilisé.

## **Problèmes à résoudre**

- synchronisation et modulation
- électronique, optique
- mécanique



**I**  
**Transmission sur un**  
**canal**

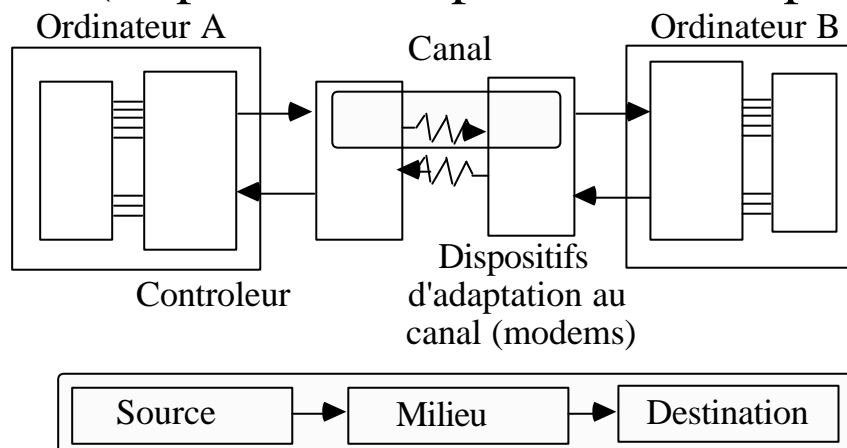
# Introduction

## Transmission sur un canal

Pour réaliser une transmission d'informations il est nécessaire de disposer d'un support physique qui véhicule les signaux électromagnétiques:

fils métalliques - signaux électriques  
 atmosphère - ondes radio  
 fibre optique - lumière

L'ensemble de transmission comporte une source (dispositif d'adaptation en émission) et une destination (dispositif d'adaptation en réception).



Un canal de transmission étudié ici est par définition unidirectionnel (ou simplexe).

## **Problème principal du niveau physique**

Quel débit d'information peut-être transmis par un canal de transmission en fonction des caractéristiques de ce canal?

En fonction de:

**- La bande passante :**

Bande des fréquences qui sont transmises par le canal.

**- La déformation du signal :**

Apportée par les imperfections de la transmission.

**- Le bruit :**

Influences externes provoquées dans le canal par le milieu extérieur.

**1**

**Transmission en bande limitée**

# Capacité de transmission d'un canal en fonction de la bande de fréquence disponible

La source utilise pour coder les données à émettre **une fonction  $g(t)$**  variable dans le temps.

Le canal est **sans bruit**.

La bande de fréquence est **limitée à une valeur  $B$** .

Outil de cette étude :  
**l'analyse de Fourier.**

## Objectif

Introduire l'importance de la disponibilité d'une large bande passante.

## Cas ou la fonction $g(t)$ est périodique

Correspond à une présentation mathématique très simplifiée.

$g(t)$  peut-être représentée comme une somme infinie de fonctions sinus ou cosinus.

$$g(t) = c_0 + \sum_{n=1}^{+\infty} c_n \cos ( 2\pi nft - \varphi_n)$$

-  $f$  la **fréquence** du signal périodique.  
 $f = 1/T$  ou  $T$  est la période.

-  $nf$  est une **harmonique** du signal.

- Chaque terme est caractérisé par

. une composante **d'amplitude**  $c_n$ .

. une composante **de phase**  $\varphi_n$

## Fonction $g(t)$ périodique: une représentation équivalente

On peut écrire également:

$$g(t) = a_0 + \sum_{n=1}^{+\infty} a_n \sin(2\pi nft) + \sum_{n=1}^{+\infty} b_n \cos(2\pi nft)$$

Calcul des différents coefficients du développement (si l'on remarque):

$$\int_0^T \sin(2\pi kft) \sin(2\pi nft) dt = \begin{cases} 0 & \text{si } k \neq n \\ T/2 & \text{si } k = n \end{cases}$$

On a:

$$a_n = \frac{2}{T} \int_0^T g(t) \sin(2\pi nft) dt$$

$$b_n = \frac{2}{T} \int_0^T g(t) \cos(2\pi nft) dt$$

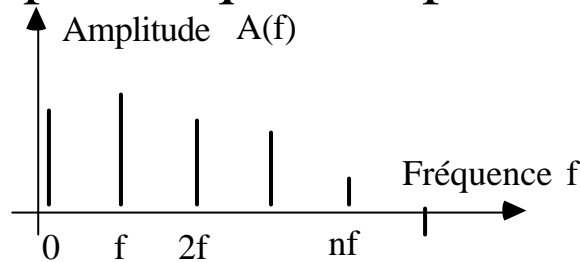
$$a_0 = \frac{1}{T} \int_0^T g(t) dt$$

# Représentation spectrale

## Spectre d'amplitude

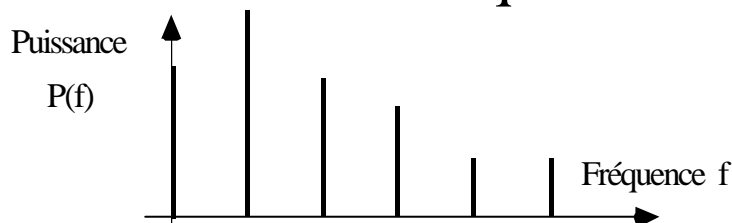
Représentation des amplitudes  $c_n$  en fonction des fréquences.

Fonction périodique  $\Rightarrow$  spectre de raies



## Spectre de puissance

Représentation des puissances contenues dans les différentes harmoniques.



Puissance moyenne d'un signal

$$P = \frac{1}{T} \int_0^T g(t)^2 dt = \sum_{n=1}^{+\infty} \frac{1}{2} (a_n^2 + b_n^2)$$

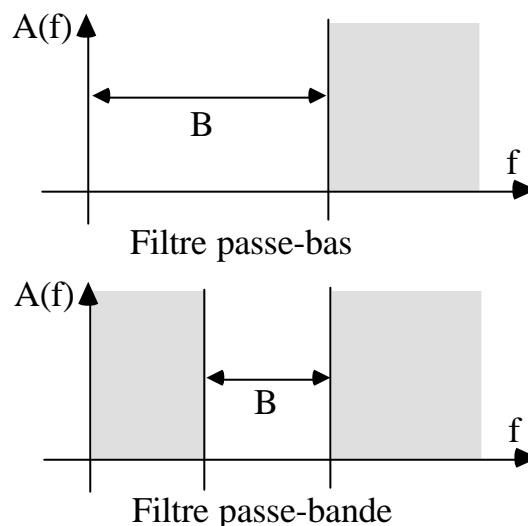


## Application de cette représentation

Lorsque l'on transmet un signal on le déforme de manière différente selon les fréquences.

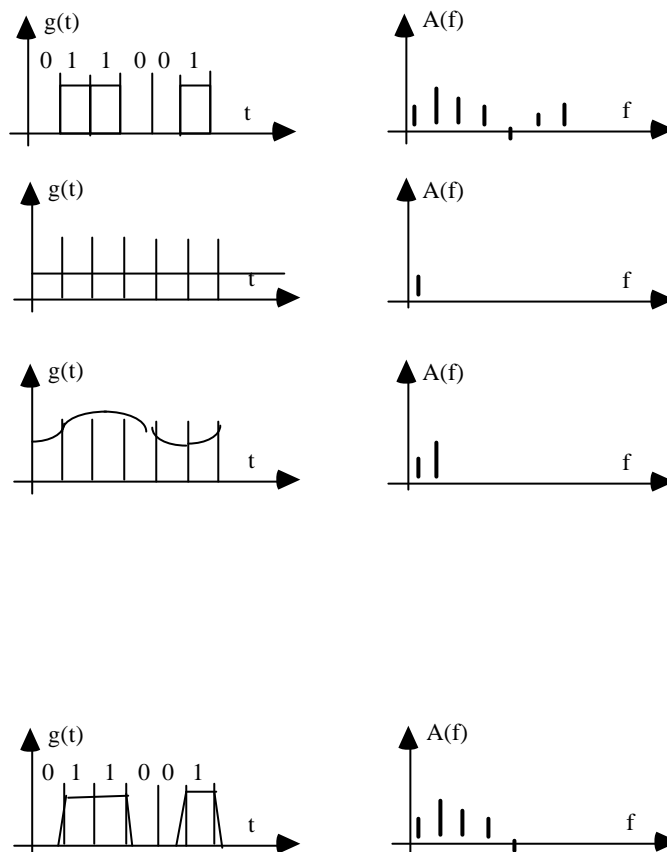
Une déformation fondamentale est qu'on ne transmet jamais toutes les fréquences.

- Les fréquences trop élevées disparaissent.



Bande passante réseau téléphonique commuté  
300-3400 Hz

# Exemple de distorsion due à la suppression des fréquences élevées



## Cas d'un signal non périodique

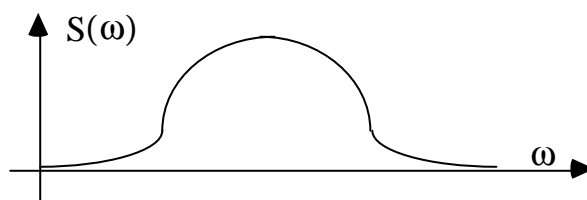
### Intégrale de Fourier

Un signal non périodique peut être mis sous la forme d'une intégrale de composantes sinusoïdales.

$$g(t) = \frac{1}{\pi} \int_0^{+\infty} S(\omega) \cos(\omega t - \Psi(\omega)) d\omega$$

### Spectre continu

Pour toutes les fréquences  $\omega$  on a une amplitude  $S(\omega)$  (et une phase  $\Psi(\omega)$  ).



## Notion de fonction de transfert

Un signal soumis à un canal par nature imparfait subit pour chaque composante:

### - Une atténuation en amplitude

Soit  $A(\omega)$  le coefficient multiplicatif qui caractérise l'atténuation en fonction de la fréquence.

### - Un retard de phase

Soit  $B(\omega)$  le coefficient additif caractérisant le retard en fonction de la fréquence.

Le signal  $g(t)$  étant émis:

$$g(t) = \frac{1}{\pi} \int_0^{+\infty} S(\omega) \cos(\omega t - \Psi(\omega)) d\omega$$

Le signal reçu est alors  $r(t)$

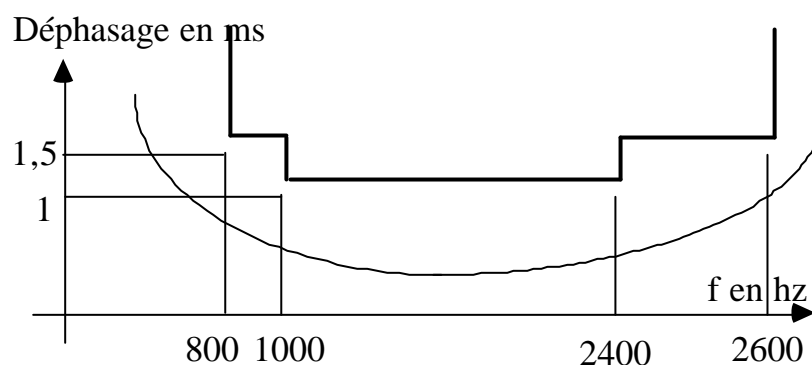
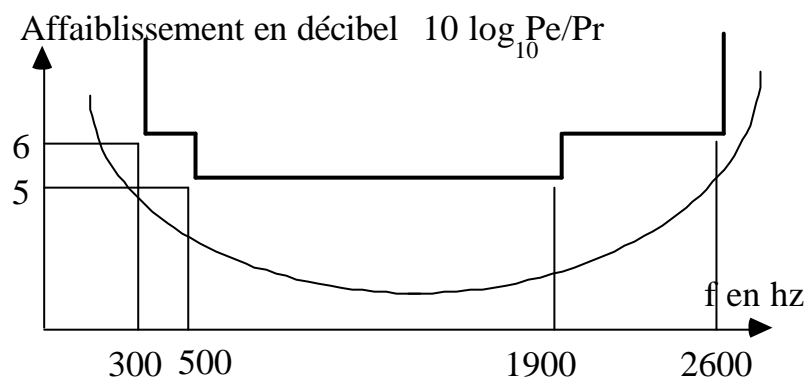
$$r(t) = \frac{1}{\pi} \int_0^{+\infty} S(\omega) A(\omega) \cos(\omega t - \Psi(\omega) + B(\omega)) d\omega$$

$A(\omega)$ ,  $B(\omega)$  Fonction de transfert caractéristique du canal.

# Engagement contractuel des prestataires de service (concernant des voies analogiques)

Performance les plus mauvaises offertes par la voie en termes d'atténuation et de déphasage:  
les gabarits.

Exemple: un gabarit de réseau téléphonique  
commuté



## Résultat de Nyquist (Gabor, Shannon)

B - La largeur de bande d'un filtre en **hertz** au travers duquel on transmet un signal

R - La rapidité de modulation en "**bauds**"  
Le nombre d'intervalles élémentaires par unité de temps (secondes) qui permettent l'échange d'informations.

V - La valence d'un signal échantillonné:  
le nombre de niveaux différents qui peuvent être distingués par intervalle.

Q - La quantité d'information par intervalle  
 $Q = \log_2 V$  en "**bits**"

C - Le débit maximum d'informations en  
**bit/seconde**:

$$C = R \log_2 V = 2 B \log_2 V$$

Pour un signal à support de largeur de bande B il ne sert à rien d'échantillonner plus de  $R = 2B$  fois par unité de temps.

Pour améliorer le débit il faut pouvoir augmenter V le nombre de niveaux.

**2**

**Transmission en présence de  
bruit**

# **Introduction à la transmission sur un canal en présence de bruit**

## **Théorie de l'information (Shannon)**

### **Objectif de l'étude**

Modéliser un canal de transmission soumis à un bruit additif.

Déterminer la capacité maximum de transmission d'un tel canal

### **Origine des bruits**

Thermiques: bruit de fond des résistances

Diaphoniques: Influence permanente d'un conducteur sur un autre

Impulsionnels : Influences transitoires des impulsions

Harmoniques : Phénomènes de battements de réflexion.



## Entropie d'une source

- Hypothèse: une source émet un message par unité de temps.

- La source sélectionne des messages (des symboles) dans un alphabet donné fini (cas infini non traité ici).

$$X = \{x_1, x_2, x_3, \dots, x_k, \dots, x_M\}$$

- Les messages émis sont aléatoires sinon il n'y a pas de communication d'information.

Ensemble des **probabilités a priori**

$$p(x_1), p(x_2), p(x_3), \dots, p(x_M)$$

- L'entropie d'une source  $H$ : la quantité d'information moyenne apportée par la source

. Quantité d'information apportée par un message précis  $-\log_2 p(x_i)$

. Quantité moyenne: espérance mathématique pour tous les messages possibles

$$H = - \sum_{i=1}^M p(x_i) \log_2 p(x_i)$$

## Influence du bruit

- Le récepteur reçoit des messages qui appartiennent à un ensemble qui n'est pas nécessairement identique à celui émis par la source.

$$Y = \{y_1, y_2, y_3, \dots, y_i, \dots, y_N\}$$

- Le bruit intervient pour modifier un message émis  $x_k$  en un message reçu  $y_i$  selon une probabilité.

Probabilité **a posteriori** (conditionnelle)

$$p(x_k / y_i)$$

Probabilité que l'émetteur ait envoyé  $x_k$  sachant que le récepteur a vu arriver  $y_i$

## Information mutuelle entre la source et le destinataire: capacité d'un canal

### Information mutuelle de deux messages émis et reçus

La quantité d'information apportée lorsqu'on reçoit  $y_i$  alors que  $x_k$  a été émis.

$$I(x_k, y_i) = \log_2 \left( \frac{p(x_k/y_i)}{p(x_k)} \right)$$

- Z - Si  $y_i$  et  $x_k$  sont indépendants  $I(x_k, y_i) = 0$ .  
- Si  $p(x_k/y_i) = 1$  on retrouve  $-\log_2(p(x_k))$

### Information mutuelle moyenne

La quantité d'information moyenne apportée au destinataire par la source

$$I(X, Y) = \sum_X \sum_Y p(x_k \text{ et } y_i) I(x_k, y_i)$$

### Capacité d'un canal

$$C = \max_{p(x)} \{ I(X, Y) \}$$

La valeur max de l'information mutuelle moyenne sur toutes les distributions a priori.

## Résultats de Shannon

### Premier résultat de Shannon

**Une source n'est caractérisée que par son entropie.**

On ne change rien sur l'information générée par la source en changeant de codage.

La seule information qui compte est son entropie (son débit en bit/unité de temps).

### Second résultat de Shannon

- **Si  $H \leq C$  il existe une codification des messages qui sur une période suffisamment longue permet de transmettre les messages avec une probabilité d'erreur résiduelle aussi faible que l'on veut.**
- **Si  $H > C$  il n'existe pas de codification qui assure sur une période de durée arbitraire une transmission sans erreurs.**

## Interprétation et conséquences

- Dans le premier cas la capacité du canal est excédentaire.

Sur une longue période cet excédent est important.

Il permet d'ajouter des redondances (sans changer l'entropie de la source) qui permettent de générer des codes correcteurs d'erreur aussi efficaces que l'on veut.

On abaisse ainsi le taux d'erreur résiduel arbitrairement.

- Dans le second cas la capacité du canal est déficitaire:

Sur une période courte on peut transmettre correctement mais ensuite on aura inévitablement des erreurs non corrigées.

- Avant ce résultat on pouvait penser que le bruit introduit une borne infranchissable sur le taux d'erreur résiduel d'une transmission.

Shannon montre que le bruit intervient sur le débit du canal et non sur sa précision.

- Pour augmenter le débit d'un canal à taux d'erreur donné on peut:

. augmenter la complexité de codage des équipements terminaux pour se placer plus près des limites du théorème.

. augmenter la capacité du canal (bande passante, puissance) avec des techniques de codage plus simples.

## Résultat particulier de Shannon

- Canal de bande passante limitée B.
- Puissance moyenne du signal S
- Puissance moyenne d'un bruit additif N.  
Bruit blanc (énergie répartie de façon uniforme dans le spectre).  
Gaussien (l'apparition d'un bruit suit une loi de gauss)

On a

$$C = B \log_2 \left( 1 + \frac{S}{N} \right)$$

Exemple:  $B = 3100 \text{ Hz}$   $10 \log_{10} S/N = 20 \text{ db}$   
 $S/N = 100$   $C = 3100 * 6,6 = 20600 \text{ b/s}$

Remarque: Dans ce cas Shannon montre que le nombre de niveaux max  $V$  qui peuvent être discriminés est donné par:

$$2B \log_2 V = B \log_2 (1 + S/N)$$

$$V = \sqrt[2]{1 + S/N}$$

# **3**

## **Détection et correction des erreurs**

# Introduction

## Distance entre deux messages

Existence de bruits qui perturbent les transmissions.

Suite binaire émise  $M$  (n uple binaire)

-----> Suite binaire reçue  $M'$

Plusieurs bits sont modifiés.

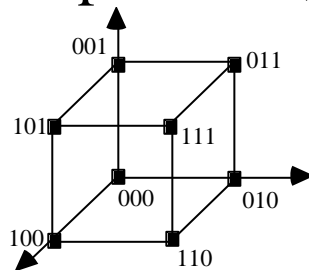
Distance entre deux messages:

distance de **Hamming**

$$d(M, M') = \sum |M(i) - M'(i)|$$

-  $d(M, M')$  est égal au nombre de bits de  $M$  et  $M'$  qui sont différents.

- C'est la distance dans l'espace à  $n$  dimensions entre les points  $M, M'$



Existence de deux stratégies très différentes de tolérance au bruit: **détection ou correction**



## Solution 1: Détection des erreurs et retransmission

Détection d'erreur par adjonction de redondances à tous les messages  $x_i$  transmis selon une fonction  $f$  connue de l'émetteur et du récepteur et retransmission si erreur.

### **Redondances "temporelles".**

A partir de l'ensemble des symboles de la source on créé un ensemble: **un code**

$$\{ x_i \} \rightarrow \{ y_i = ( x_i , f(x_i) ) \}$$

Symboles à émettre  $\rightarrow$  Mots du code émis

Un message reçu  $y_j = ( x_j , z_j )$  est correct (appartient au code) si  $z_j = f(x_j)$ .

Il n'appartient pas au code si  $z_j$  non égal à  $f(x_j)$ .

D la distance des messages les plus proches  
(**la distance du code**):

Le code est détecteur de  $D-1$  erreurs.

Si un message est détecté incorrect

$\Rightarrow$  Retransmission ultérieure.

## Solution 2: Codes auto-correcteurs d'erreurs

- Constitution d'un code par ajout de redondances comme précédemment.

$$\{ x_i \} \rightarrow \{ y_i = ( x_i , f(x_i) ) \}$$

- Si le message reçu appartient au code on l'accepte.

- Si le message reçu est erroné

$$y_j = ( x_j , z_j ) \text{ avec } z_j \text{ différent de } f(x_j)$$

on fait l'hypothèse que le bruit a peu altéré le message codé.

- On corrige un message erroné par le mot du code correct le plus proche du mot de code erroné reçu.

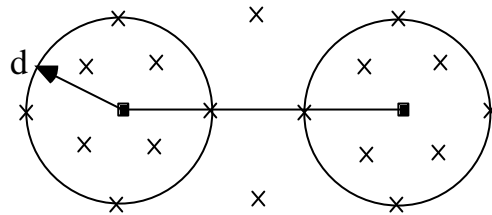
$$( x_i , f(x_i) ) = \inf_k [d((x_j , z_j), (x_k, f(x_k)))]$$

### **Redondances "spatiales" pour le masquage des erreurs.**

Remarque: On ne corrige pas toutes les erreurs. Si le poids de l'erreur est trop élevé:  
=> erreur de correction.

## Correction automatique: représentation géométrique des messages

- On trace des sphères centrées sur chaque mot d'un code (les messages corrects)
- Rayon  $d$  (distance de Hamming).



messages corrects: carrés noirs  
messages incorrects : croix.

### Hypothèse importante pour la correction

Les erreurs de faible poids (portant sur un petit nombre de bits) ont une probabilité d'apparition très forte par rapport aux erreurs de fort poids.

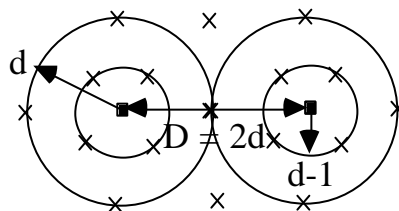
Pour un  $d$  donné un erreur va conduire avec une probabilité élevée à un message erroné qui se trouve à l'intérieur de la sphère de rayon  $d$  (probabilité faible à l'extérieur).

## Principes de la correction automatique

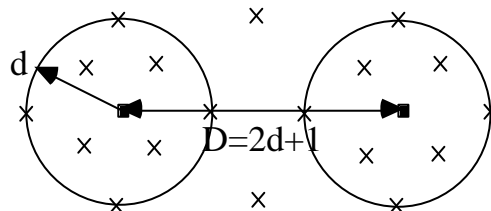
Si les sphères de rayon  $d$  ayant pour centre les messages corrects ont une intersection vide on peut **corriger tout message incorrect** dans l'une des sphères **par le message correct au centre** ( $D$  distance des messages les plus proches **distance du code**):

- On peut détecter les erreurs de poids  $D-1$ .

Si  $D=2d$  correction des erreurs de poids  $d-1$



Si  $D=2d+1$  correction des erreurs de poids  $d$



**Problème de construction d'un code correcteur:** écartier le plus possible et de façon régulière les mots du code par le choix de la fonction  $f$  (redondance).

# Paramètres d'un code

## a) Taux d'erreur brut

$$\tau = \frac{\text{Nombre de messages faux}}{\text{Nombre de messages total}}$$

## b) Efficacité d'un code

$$e = \frac{\text{Nombre de messages reconnus faux}}{\text{Nombre de messages faux total}}$$

## c) Taux d'erreur global résiduel

$$q = \frac{\text{Nombre de messages finalement faux (erreur non détectée, non corrigée)}}{\text{Nombre de messages total}}$$

## b) Rendement d'un code

$$r = \frac{\text{Nombre de bits utiles reçus}}{\text{Nombre de bits envoyés}}$$

## Codes linéaires

Mots du code: ensemble de n-uples binaires (vecteurs de n bits) formant un espace vectoriel sur  $(0,1, \oplus, \cdot)$

$\oplus$	0	1
0	0	1
1	1	0

$\cdot$	0	1
0	0	0
1	0	1

- Si  $X_1$  et  $X_2$  appartiennent au code  
 $X_1 \oplus X_2$  appartient au code.
- 0 appartient au code.

## Matrice de génération

G (k lignes, n colonnes) matrice qui fait correspondre à un message Y de k bits un mot du code X de n bits.

G ajoute une redondance  $m = n - k$  bits.

$$X = Y \cdot G$$

$$(1, n) = (1, k) (k, n)$$

Exemple: le contrôle de parité simple

$$(x_1, \dots, x_k, x_{k+1}) = (x_1, \dots, x_k) \begin{bmatrix} 1 & \dots & 0 & 1 \\ & \diagdown & & 1 \\ & & 1 & 1 \\ & & & \diagdown \\ 0 & \dots & & 1 & 1 \end{bmatrix}$$

$$x_{k+1} = x_1 \oplus x_2 \oplus \dots \oplus x_k$$

## Codes séparés

Bits d'informations, bits de contrôle.

G est de la forme  $G = [I_{k,k} P_{k,n-k}]$

$$X = (y_1, y_2, \dots, y_k, x_{k+1}, \dots, x_n)$$

(y<sub>1</sub>, y<sub>2</sub>, ..., y<sub>k</sub>) Message utile

(x<sub>k+1</sub>, ..., x<sub>n</sub>) Bits de contrôle

## Matrice de contrôle

- $V'$  l'espace **orthogonal** de  $V$
- Tout mot du code  $X \in V$  a un **produit scalaire nul** avec tout vecteur  $X' \in V'$ .
- Tout message  **$X'$  en erreur** (n'appartenant pas au code) **appartient à  $V'$** .
- Tout message  $X'$  (en erreur) a un produit scalaire non nul avec au moins un vecteur  $\in V'$
- Soit  **$H$  ( $n, n-k$ )** la matrice génératrice de l'orthogonal (la matrice formée au moyen des vecteurs d'une base de  $V'$ ).
- **Syndrome** d'un message:  $S = X H$
- . Si  **$S = \mathbf{0}$**  le message est présumé **correct**
- . Si  **$S$  différent de  $\mathbf{0}$**  le message est **erroné**.



## Construction de la matrice de contrôle

Très facile pour les codes séparés

On vérifie que si  $G = [I_{k,k} P_{k,n-k}]$

$$H = \begin{bmatrix} P_{k,n-k} \\ I_{n-k,n-k} \end{bmatrix}$$

Alors

Si l'on utilise la transposée de H

$$H^T = [P_{n-k,k}^T I_{n-k,n-k}]$$

### Cas du contrôle de parité simple

$$G = \begin{bmatrix} 1 & \text{---} & 0 & 1 \\ & \diagdown & & 1 \\ & & 1 & \vdots \\ 0 & \text{---} & & 1 \end{bmatrix}$$

$$H^T = [1 \ 1 \ \dots \ 1 \ 1 \ 1]$$

$$S = (x_1, \dots, x_k, x_{k+1})H$$

$$S = x_1 \oplus x_2 \oplus \dots \oplus x_k \oplus x_{k+1}$$

## Codes polynomiaux

- A un message on associe un polynôme.

Ex: 1 1 0 1 1 1 -->  $x^5 \oplus x^4 \oplus x^2 \oplus x \oplus 1$

On veut ajouter deux bits de redondance: on multiplie par  $x^2$  -->  $x^7 \oplus x^6 \oplus x^4 \oplus x^3 \oplus x^2$

- On choisit un polynôme générateur de degré inférieur au degré du polynôme message.

Exemple :  $x^2 \oplus x \oplus 1$

- Les bits de redondance ajoutés au message sont les coefficients du polynôme reste dans la division euclidienne du polynôme message par le polynôme générateur.

$$\begin{array}{r|l}
 x^7 \oplus x^6 \oplus x^4 \oplus x^3 \oplus x^2 & x^2 \oplus x \oplus 1 \\
 \underline{x^7 \oplus x^6 \oplus x^5} & \hline
 x^5 \oplus x^4 \oplus x^3 \oplus x^2 & x^5 \oplus x^3 \oplus 1 \\
 \underline{x^5 \oplus x^4 \oplus x^3} & \\
 x^2 & \\
 \underline{x^2 \oplus x \oplus 1} & \\
 x \oplus 1 & 
 \end{array}$$

Le message transmis: 1 1 0 1 1 1 1 1

## Codes polynomiaux Présentation formelle

- Message de  $k$  bits :  $M$
- Polynôme associé au message  $M(x)$  de degré  $k-1$ . On décale de  $m$  positions.

En fait on utilise le polynôme  $x^m M(x)$

- $G(x)$  le polynôme générateur du code de degré  $m$ .

- On effectue la division:

$$x^m M(x) = G(x) Q(x) \oplus R(x)$$

- On obtient un reste  $R(x)$  de degré  $m-1$  au plus qui a la propriété suivante:

$$x^m M(x) \oplus R(x) = G(x) Q(x)$$

- L'ensemble des polynômes (degré  $m+k-1$ ) tels que  $C(x) = x^m M(x) \oplus R(x)$

forment **un code polynomial**.

Tout **mot du code** donne un **reste nul** dans la division par  $G(x)$ .

Tout **mot hors du code** (message erroné) **donne un reste non nul**.

## Propriétés des codes polynomiaux

**- Un code polynomial qui génère m bits de redondance détecte toutes les rafales d'erreurs de longueur < m.**

Une rafale de longueur k se présente comme une erreur additive de la forme

$$E(x) = x^i (x^{k-1} \oplus x^{k-2} \oplus \dots \oplus x^2 \oplus x \oplus 1)$$

Pour qu'une telle erreur soit indétectée il faut que E(x) soit divisible par G(x).

Si le polynôme G(x) a un terme constant il ne peut pas avoir x<sup>i</sup> en facteur.

Si k-1 est plus petit ou égal à m-1 le degré de  $(x^{k-1} \oplus x^{k-2} \oplus \dots \oplus x^2 \oplus x \oplus 1)$  est inférieur au degré de G(x) et G(x) ne peut donc le diviser.

**- Un code polynomial détecte toutes les erreurs simples.**

Une erreur simple se présente comme une erreur additive de la forme:  $E(x) = x^i$

Pour qu'une telle erreur soit indétectée il faut que E(x) soit divisible par G(x).

Si G(x) a plus d'un seul terme il ne peut être diviseur de E(x)

**- Nombreuses autres propriétés.**

## Exemples de codes polynomiaux

### CRC-12

Définition d'un polynôme générant 12 bits de redondance

$$G(x) = x^{12} \oplus x^{11} \oplus x^3 \oplus x^2 \oplus x \oplus 1$$

### Avis V41 CCITT

Définition d'un polynôme CRC-CCITT  
(protocoles de liaisons en point à point dérivés de HDLC)

$$G(x) = x^{16} \oplus x^{12} \oplus x^5 \oplus 1$$

### CRC-IEEE 802

Définition d'un polynôme générant 32 bits de redondance

(Ethernet, boucle IBM FDDI)

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

# 4

## **La représentation des signaux**

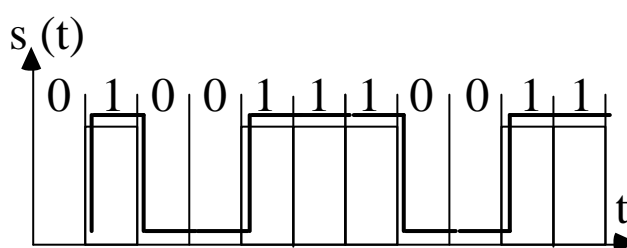
# Introduction

On s'intéresse à une transmission série

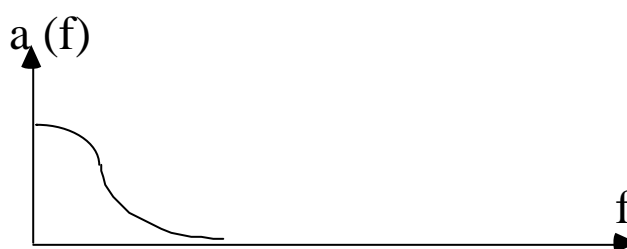
## Transmission en bande de base

Le signal est dans une représentation numérique en fonction du temps sorti d'un système informatique.

Exemple type: le codage NRZ-L



Le spectre associé.

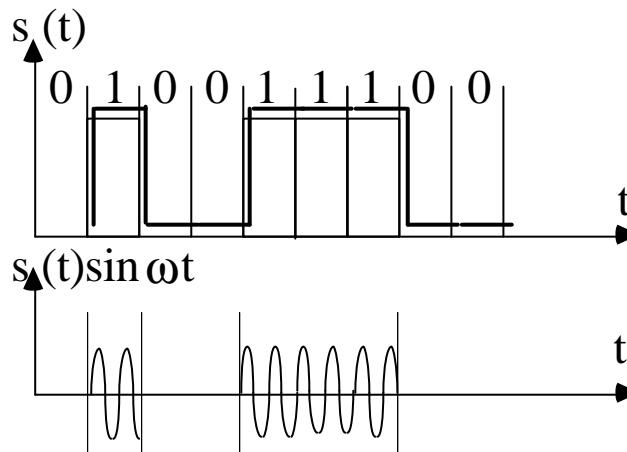


## Transmission en modulation (d'onde porteuse)

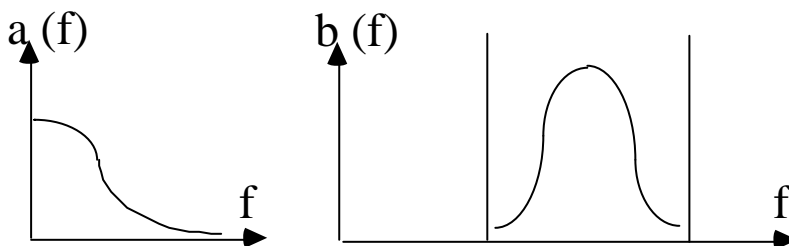
Le signal  $s(t)$  est représenté au moyen d'une onde porteuse qui modifie le codage de base pour s'adapter à un canal de transmission.

- positionnement dans le spectre utile
- remise en forme pour occuper la bande disponible.

Exemple type: la modulation d'amplitude



Le spectre associé est modifié.





## **Problèmes de synchronisation dans les transmissions de données numériques**

Le récepteur d'un signal doit connaître la position de chaque bit pour échantillonner correctement les valeurs.

Nombreuses difficultés:

- Détermination du début des suites binaires significatives (notion de trames de bit)
- Echantillonnage des bits en présence de multiples aléas de fonctionnement (gigue, dérive entre les horloges, délais de propagation,...).

## Terminologie concernant les signaux

### Signaux Isochrones (égaux)

Il existe **un écart fixe** entre deux signaux successifs. Exemples:

**Son** : le réseau téléphonique utilise des échantillons de 8 bits isochrones selon une base de temps de 125 microseconde.

**Image** : la vidéo utilise généralement des images de format donné espacées de 40 millisecondes.

**Z** L'intervalle constant doit être reproduit fidèlement chez le récepteur sous peine de perte de qualité de la restitution.

### Signaux anisochrones (Antonyme du précédent)

Il n'y a **pas d'intervalle fixe** entre les signaux. Il peut néanmoins être très important de **restituer l'espacement variable** de l'émission lors de la délivrance au récepteur (contraintes temps réel).

## **Signaux synchrones (ensembles)**

Des signaux synchrones sont à la même cadence (sont rythmés par la **même horloge**).

## **Signaux asynchrones**

Antonyme du précédent

Des signaux asynchrones n'apparaissent pas selon un rythme constant défini par une horloge mais apparaissent **aléatoirement**.

## **Signaux plésiochrones (voisins)**

Des signaux plésiochrones sont rythmés par des horloges dont les **fréquences** sont **peu différentes** (plésio = voisin).

## **Signaux Mésochrones (moyens)**

Les écarts entre signaux ne sont pas constants mais la **moyenne des intervalles est fixe** (le débit est constant).

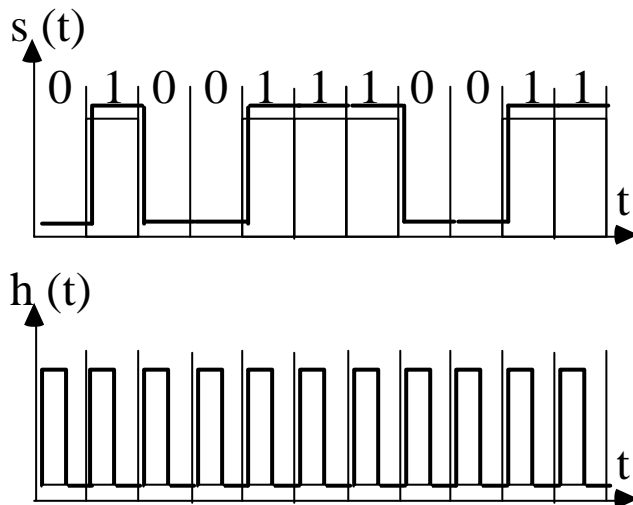
Permet de restituer des signaux isochrones par utilisation de mémoire tampon (par exemple en ATM).

# Terminologie concernant les transmissions

## Transmission synchrone

**On transmet une horloge** sur un canal spécial de l'émetteur au destinataire de sorte que ces deux sites peuvent utiliser exactement la même base de temps => génération et échantillonnage selon le même rythme.

Il faut une bonne qualité d'acheminement de l'horloge => **Solution assez coûteuse** en bande passante nécessitant un canal spécial pour l'horloge.



## Transmission asynchrone (Start-Stop)

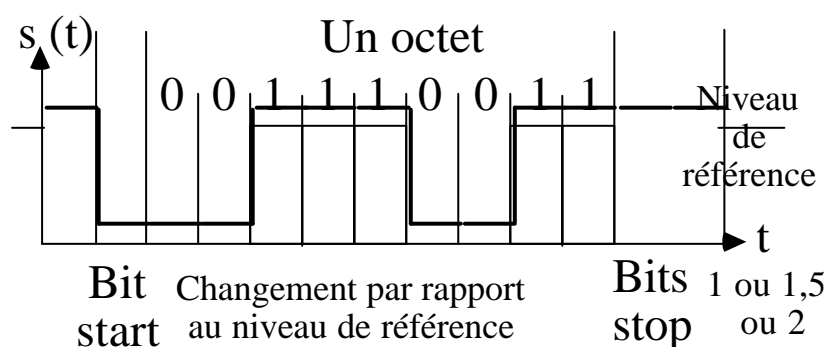
Pas d'horloge commune.

L'asynchronisme se place sur l'instant de **commencement d'une suite binaire.**

On ne transmet que des suites binaires courtes **isochrones** (en fait des octets).

On table sur une dérive relative entre l'horloge d'émission et l'horloge de réception qui permet de ne pas perdre la synchronisation bit.

Adapté à des **débits faibles.**



## Transmission plésiochrone

Les horloges émetteur et destinataire sont **différentes mais voisines** (avec une **tolérance** liée au débit visé).

On synchronise une horloge de réception sur l'horloge d'émission.

Exemple: usage d'un préambule présent dans tout message (typiquement un signal d'horloge).

Le message est ensuite échantillonné correctement par le récepteur mais il peut être selon les dérives relative des horloges plus ou moins long

=> Nécessité de tenir compte de cette longueur pour le stockage:

- registres d'élasticité
- mémoires tampons.

=> Adaptation aux vitesses relatives de horloges (techniques de justification):

=> Nécessité de déterminer la position des informations significatives (pointeurs).

# Techniques de codage en bande de base

## Signaux bande de base en amplitude

### Les codages NRZ

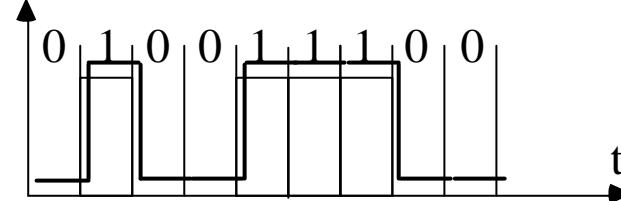
**"Non Retour à Zéro"**: Le niveau est constant sur un intervalle (il n'y a pas de transition de retour à zéro)

**NRZ-L** ("Level") On utilise deux niveaux pour coder le 0 et le 1. **Exemple: V24.**

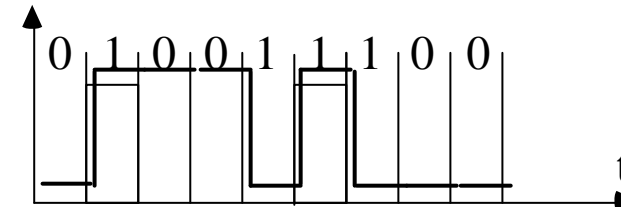
**NRZI** ("Inverted") Codage différentiel des 1. Chaque nouveau 1 entraîne un changement de polarité par rapport au précédent 1 alors que le 0 n'entraîne pas de modification

Exemple: FDDI.

NRZ-L



NRZI



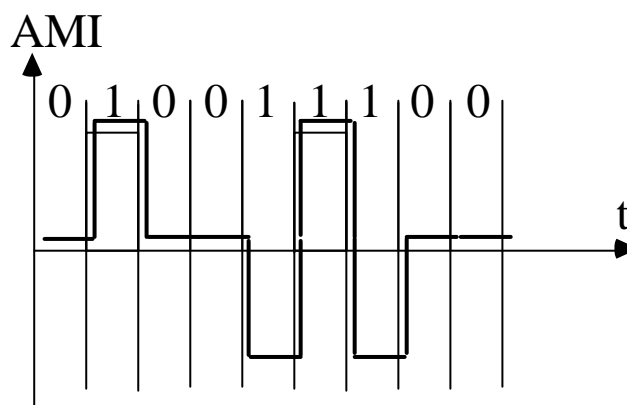
## Les codages AMI et pseudoternaire

On utilise trois niveaux l'un à 0 , +V, -V volts (code bipolaire).

**Bipolaire AMI:** "Alternate Mark Inversion"  
Un 0 est représenté par le signal 0 volts et un 1 est représenté alternativement par +V et -V volts

**Exemple: RNIS Bus d'abonné.**

**Pseudo ternaire :** On inverse le 0 et le 1 dans le schéma précédent.



**Avantages:**

Bonne occupation de la bande passante.

On peut resynchroniser sur les séquences de 1 (pas sur les 0).

Détection d'erreur sur les ajouts de transition.



## **Techniques d'embrouillage ("Scrambling")**

Problème des codages NRZ ou AMI  
Absence de transitions dans certaines longues séquences de symboles identiques.

=> Ajouter une technique qui permet de forcer l'apparition de transitions dans ces séquences.

### **Embrouillage par utilisation d'un code polynomial**

Technique employée:

- Ou exclusif avec une séquence pseudo aléatoire générée par un polynôme

La séquence pseudo-aléatoire comporte des 0 et des 1 uniformément distribués

=> ils apparaissent dans le message transmis.

Opération identique à l'arrivée.

#### **Exemple: Réseau ATM (cellules)**

"Asynchronous Transfer Mode"

- Polynôme  $X^{31} + X^{28} + 1$

## Codages HDB3 et B8ZS

Code de base AMI. On remplace des suites binaires de taille fixe (4, 8) sans transitions.

### **B8 ZS "Bipolar with eight zero substitution"**

Si 8 bits consécutifs sont à 0 on les remplace. On génère ainsi une violation.

- Si la dernière excursion était positive (+) on remplace par la chaîne 000+-0-+ .

- Si la dernière excursion était négative (-) on remplace par la chaîne 000-+0+- .

**Exemple: RNIS américain.**

### **HDB3 "High Density Bipolar - three zero"**

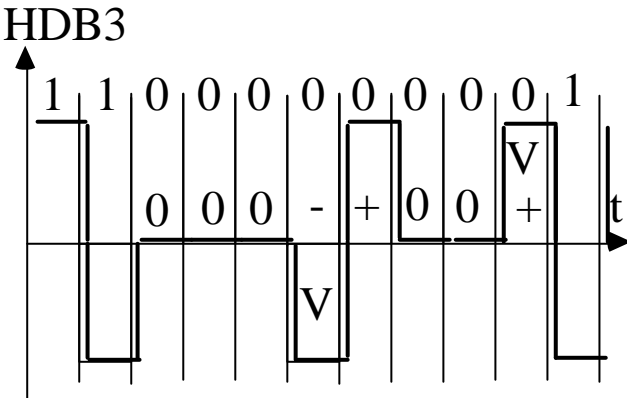
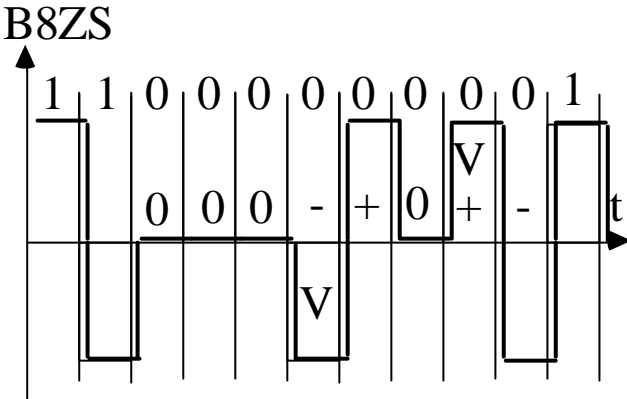
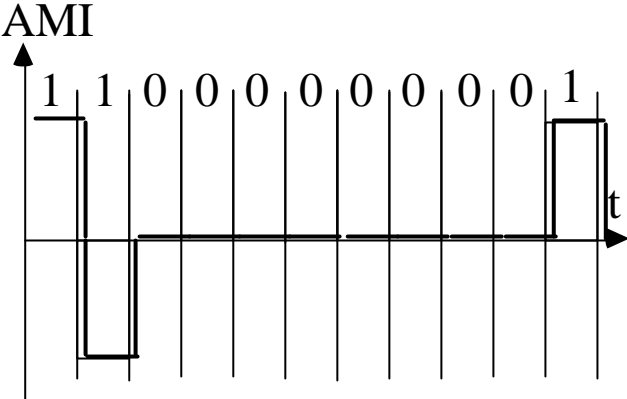
Si 4 bits consécutifs sont à 0 on les remplace selon une règle qui dépend de la polarité de la dernière excursion et également du nombre d'excursions.

Polarité précédente: Nombre d'excursions  
depuis la dernière violation.

	Impair	Pair
-	000 -	+00+
+	000+	- 00-

**Exemple: RNIS européen.**

# Exemples HDB3 et B8ZS



## Codages nB/mB

On représente n bits usagers sur m bits  $n < m$

On peut ainsi choisir des configurations qui présentent toujours un nombre suffisant de transitions.

### **Exemple: Réseau FDDI**

("Fiber Distributed Data Interface")

Codage 4 bits usagers sur 5 bits transmis

=> On choisit des configurations binaires telles qu'**il existe toujours au moins une transition par groupe de trois bits.**

=> Les autres configurations sont utilisées pour **la signalisation** (délimiteurs de trames, acquittements en fin de trame...)

### **Exemple: Réseau FC ("Fiber Channel")**

Débit 1 GigaBit/S

Codage 8B/10B

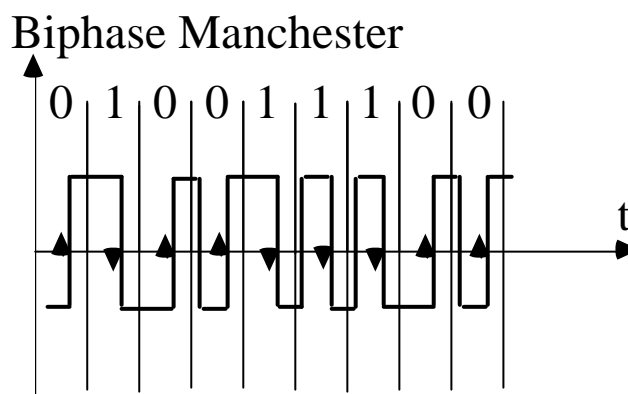
## La transmission en bande de base en phase

### Codage biphasé (code manchester)

Le signal présente toujours une transition au milieu de l'intervalle:

- pour coder un 0 transition vers le haut
- pour coder un 1 transition vers le bas.

**Exemple: Ethernet.**



**Avantages:** Pas de composante continue.

Bonne occupation de la bande passante.

Resynchronisation d'horloge sur les transitions en milieu de période.

Détection d'erreur sur l'absence d'une transition prévisible.

**Inconvénients:** Nécessite une bande passante très large.

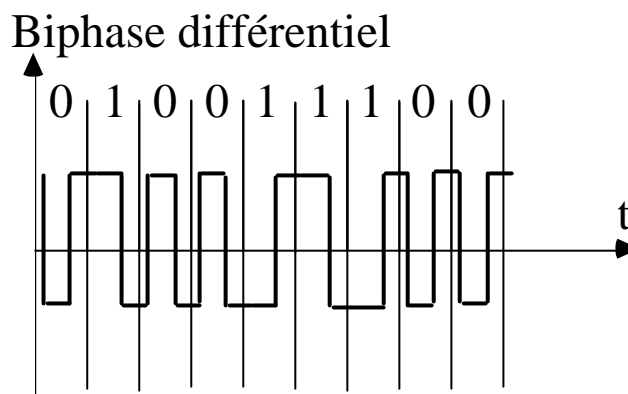
## Codage biphasé différentiel ("Differential Manchester")

Le signal présente toujours une transition au milieu de l'intervalle qui sert pour la resynchronisation d'horloge.

Codage du 0: transition en début de période

Codage du 1: absence de transition.

**Exemple: boucle IBM IEEE 802.5.**



**Avantages Inconvénients:** Voisins de ceux du code Manchester.

Usage d'une signalisation différentielle.

## **Autres techniques de codage en bande de base**

### **Modulation d'impulsion**

Codage RZ: le signal est sous forme d'impulsions (présente toujours un retour à zéro).

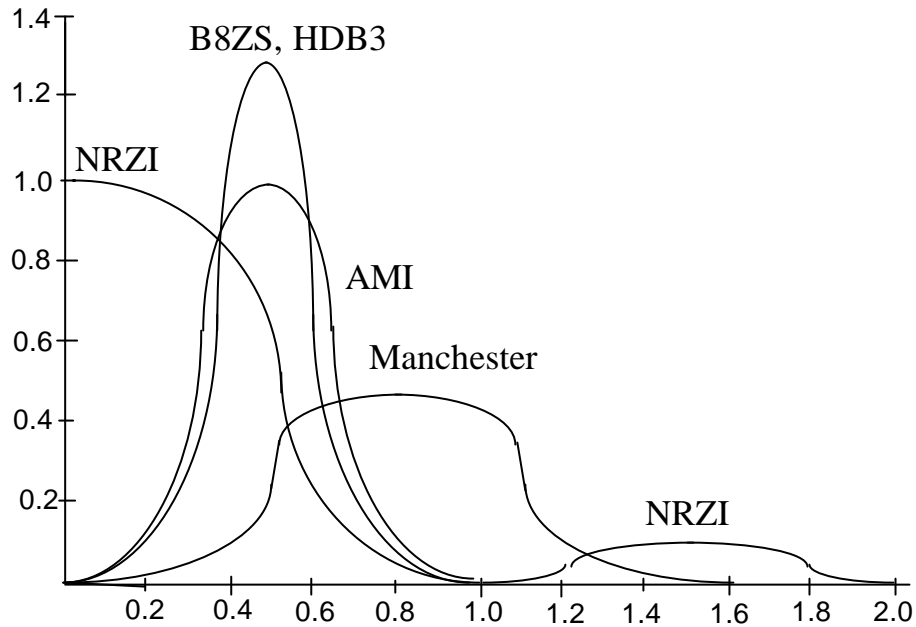
Exemple: Morse modulation d'impulsion en durée.

### **Modulation de fréquence**

Deux fréquences d'horloge différentes sont utilisées pour coder le 0 et le 1.

Améliorations: suppression de transitions.

## Comparaison des différentes techniques



Puissance par unité de bande passante en fonction de la fréquence normalisée.



## **La transmission analogique Modulation d'onde porteuse - Modems**

- Limitation des bandes de fréquence disponibles

Exemple : Téléphone 300 à 3400 Hertz.

- Affaiblissement et déformation des signaux dues aux caractéristiques des câbles, longueur, etc....

=> Nécessité de dispositifs d'adaptation de la source/destinataire au canal: les modems.

Terminologie UIT-CCITT les ETCD

ETCD : Équipement de terminaison de circuit de données.

## Transmission en modulation Principe général

Signal de base  $S(t)$

Porteuse sinusoidale  $P(t)=A_o \sin (\omega_o t + \phi_o)$

Modulation d'onde porteuse: on transforme  $S(t)$   
->  $X(t) = f(S(t))$  en introduisant l'information  
du signal dans l'une des composantes:

Amplitude  $A$

Fréquence  $\omega$

Phase  $\phi$

## Modulation d'amplitude

$X(t) = f(S(t)) = A(S(t)) \sin (\omega_o t + \phi_o)$

Exemple de base: Deux amplitudes

0  $A_1 \sin (\omega_o t + \phi_o)$

1  $A_2 \sin (\omega_o t + \phi_o)$

## Modulation de fréquence

$$X(t) = f(S(t)) = A_o \sin (\omega(S(t)t + \phi_o)$$

Exemple de base: Utilisation de deux fréquences FSK "Frequency Shift keying"

$$0 \quad A_o \sin (\omega_1 t + \phi_o)$$

$$1 \quad A_o \sin (\omega_2 t + \phi_o)$$

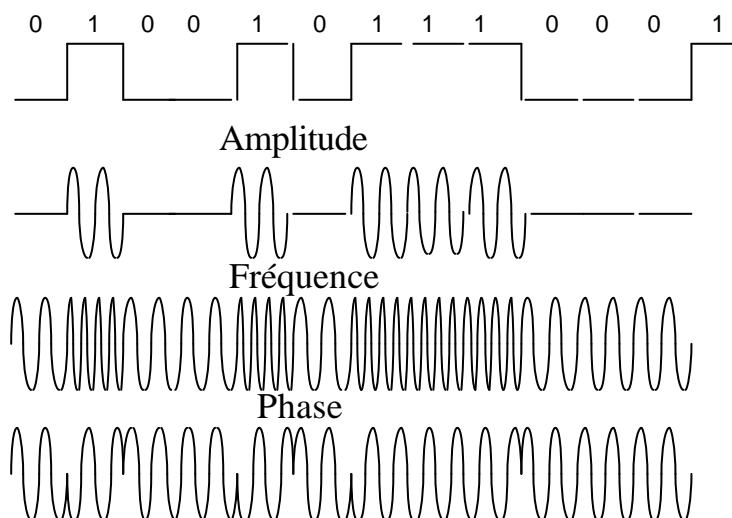
## Modulation de phase

$$X(t) = f(S(t)) = A_o \sin (\omega(S(t)t + \phi_o)$$

Exemple de base: Utilisation de deux phases PSK "Phase Shift keying"

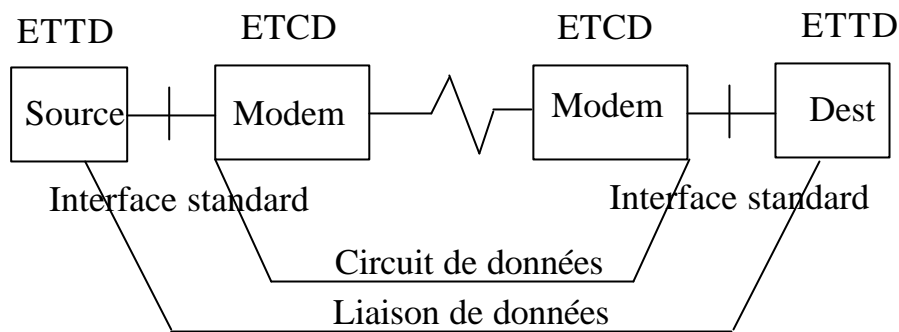
$$0 \quad A_o \sin (\omega_o t + \phi_1)$$

$$1 \quad A_o \sin (\omega_o t + \phi_2)$$



**Les éléments de technologie  
du niveau physique**

# Introduction



ETTD : Équipement terminal de traitement des données

ETCD : Équipement de terminaison de circuit de données.

## Différents éléments intervenant dans la transmission

Contrôleurs de communication

Interfaces standards

Voies de communication

Modems

## Les contrôleurs de communication

- Acquisition des informations en mémoire:  
mode Direct Memory Access
- Transmission en émission  
Gestion des délimiteurs/Sérialisation  
Génération des codes détecteurs
- Transmission en réception  
Gestion des délimiteurs/Désérialisation  
Gestion de mémoire tampon  
Registres à décalages  
Mémoire locale contrôleur  
Vérification des codes détecteurs.
- Gestion de l'accès au médium  
Cas des réseaux locaux.
- Administration/configuration  
Adaptation aux différents débits  
Adaptation aux différents modes  
(synchrone, asynchrone)  
Gestion des mots d'état.

## **Les interfaces normalisées**

L'ensemble de spécifications mécaniques et des utilisations des signaux électriques permettant une normalisation "relative" des interfaces ETTD-ETCD.

### **L'interface V24/V28 ou RS 232C**

L'avis V24 du CCITT définit les spécifications fonctionnelles de la jonction; V28 correspond aux caractéristiques électriques des signaux de la jonction.

Il existe quelques différences mineures entre les interfaces V24 et RS232C.

Les protocoles sont codés par des niveaux ouverts/fermés sur des signaux (nomenclature série 100) correspondant à des broches du connecteur DB25.

Exemple: 125 (22)

#### **Trois étapes principales**

- Établissement / libération du circuit
- Initialisation
- Transmission

## **Établissement et libération du circuit**

- 125 (22): Indication d'appel (coté modem), (ex : sonnerie du téléphone) ETCD -> ETTD.
- 108/1 (20): Connexion sans condition
- 108/2 (20): Autorisation de connexion (du modem) ETTD->ETCD.
- 107 (06): Acquiescement indiquant que le modem est relié à la ligne ETCD -> ETTD.

## **Initialisation**

- 105 (04): Demande pour émettre ETTD->
- 106 (05): Prêt à émettre ETCD -> ETTD
- 109 (08): Détection de porteuse ETCD ->

## **Transmission**

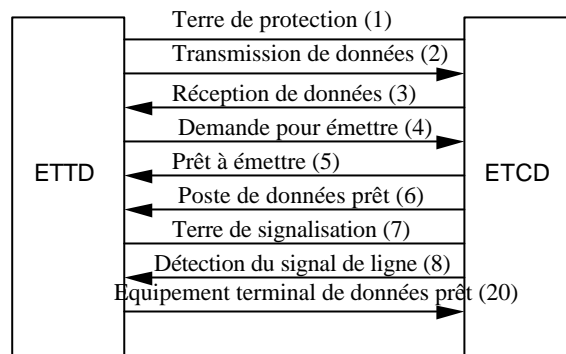
- 103 (02): Données ETTD vers ETCD
- 104 (03): Données ETCD vers ETTD
- 113 (24): Horloge émission (terminal pilote)
- 114 (16): Horloge émission (modem pilote)

## **Divers**

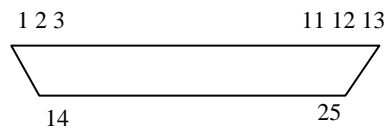
- 102 (07): Terre de signalisation
- 101 (01): Terre de protection
- 111 (23): Sélection de débit binaire



## Représentation logique de l'interface



## Connecteur V24



ETTD (Terminal)

## Normes électriques V28

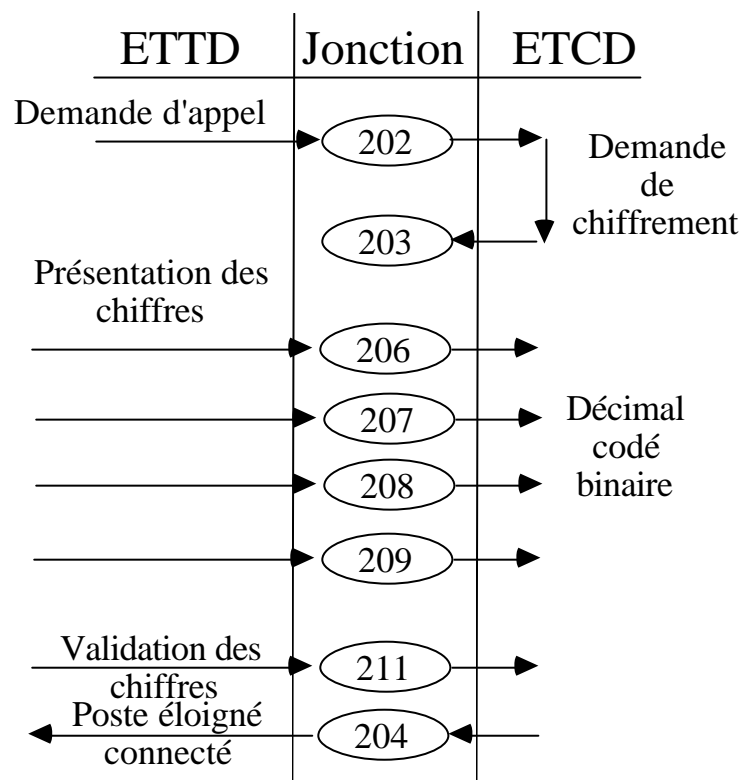
(-3V) à (-25V)      1 logique  
(+4V) à (+25V)      0 logique  
on prend -5,+5 v

## V24 et les normes associées

- **RS 232 C** : EIA (RS 232 C = V24 + V28).
  
- **RS 449** : Modernisation de RS 232  
Deux normes RS 422, RS 423
  - . Ajout de fonctions (test des modems locaux et distants)
  - . Modification de l'interface électrique pour améliorer les débits et la portée (V24 est limité à 19200 b/s et 15 m).
  
- **RS 423-A** ou V10: amélioration de RS 232 avec signalisation sur un seul fil débit jusqu'à 100 kb/s - portée 1000m.
  
- **RS 422-A** ou V11 : amélioration de RS 232 avec signalisation sur deux fils débit 2-10 Mb/s - portée 60-1000m.
  
- **V35** : Une norme de modem à 48 kb/s en modulation d'amplitude à bande latérale unique pour lequel en l'absence de standard à l'époque on a spécifié une interface qui est utilisée pour les débits au dessus de 19200 b/s => Connecteur spécifique 34 broches.

## La prise de contrôle de ligne Standards V25 et V25bis

Objectif réaliser les fonctions de gestion du réseau téléphonique commuté (numérotation)  
=> Signaux série 200.



V25 bis réduit le nombre de signaux de la série 200 pour acheminer le minimum indispensable sur la prise DB25.

## **La prise de contrôle de ligne Standard Hayes**

Objectif: établir un dialogue en ASCII avec un modem pour transmettre des commandes de configuration, de numérotation.

### **Deux modes de fonctionnement**

- Lorsqu'aucune connexion n'est établie un modem Hayes est un appareil qui interprète tout ce qu'il reçoit comme des commandes de configuration.

Intérêt: aucun logiciel spécifique n'est nécessaire, tout peut être tapé au clavier.

- Lorsqu'une connexion est établie toutes les informations transmises au modem sont envoyées à l'équipement distant.

### **Exemple de commandes**

Format AT ..... Carriage Return

Réponses OK, ERROR, CONNECT ....

ATA : décrocher

ATDn : composer le numéro de téléphone n

ATE0/1:0 pas d'écho de commande/1 écho.

## **Autres interfaces de niveau physique**

- Interface S  
Voir chapitre RNIS

- Interfaces normalisées  
des réseaux locaux

Exemple interface AUI ethernet  
Voir chapitre réseaux locaux.

## **Les modems (modulateurs/démodulateurs)**

**1) Modems téléphoniques** (utilisation du téléphone fixe sur paire téléphonique).

- **Modems classiques (débits faibles)**
- **Modems ADSL**

**‘Asymmetric Digital Subscriber Line’**

**2) Modems cables**

**HFC ‘Hybrid Fiber Coaxial Cable’**

( Utilisation des câbles de télédistribution)

## Modems téléphoniques

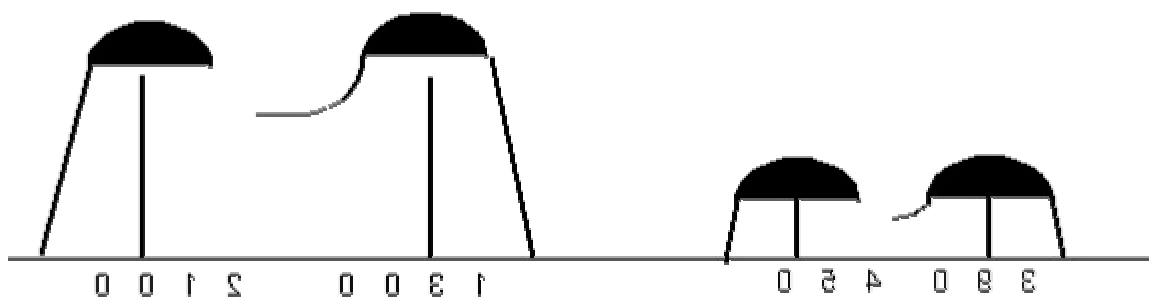
Exemples présentés V23, V32, V90, ADSL.

### Modem Avis V23

- Modem du minitel
- Débit : 600/1200 bits/seconde
- Transmission : asynchrone
- Support utilisable : Réseau commuté (2 fils) ou Lignes spécialisées (4 fils)
- Mode : Duplex à l'alternat (ancien)  
Duplex intégral (600b/s ou 1200 b/s  
voie de retour 75b/s)

- Principe : modulation de fréquence

Deux canaux : Descendant		Retour	
1	2100 Hz	390 Hz	
0	1300 Hz	450 Hz	



## Modem Avis V32

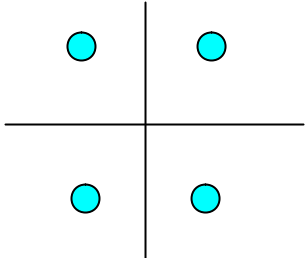
- Débit : 2400,4800,9600 bits/seconde.
  - Transmission : synchrone.
  - Support utilisable : 2 fils ou RTC.
  - Mode : duplex
  - QAM (Quadrature Amplitude Modulation) Modulation d'amplitude et de phase en quadrature.
  - Débit    Rapidité de modulation    Valence  
9600b/s        2400 bauds        32
- Le débit réel 12 000 b/s est supérieur au débit utile permettant l'utilisation d'un code auto-correcteur implanté par le modem.



# Les modulations d'amplitude et de phase

Exemple 1 : Modulation de phase (QPSK Quadrature Phase Shift Keying)

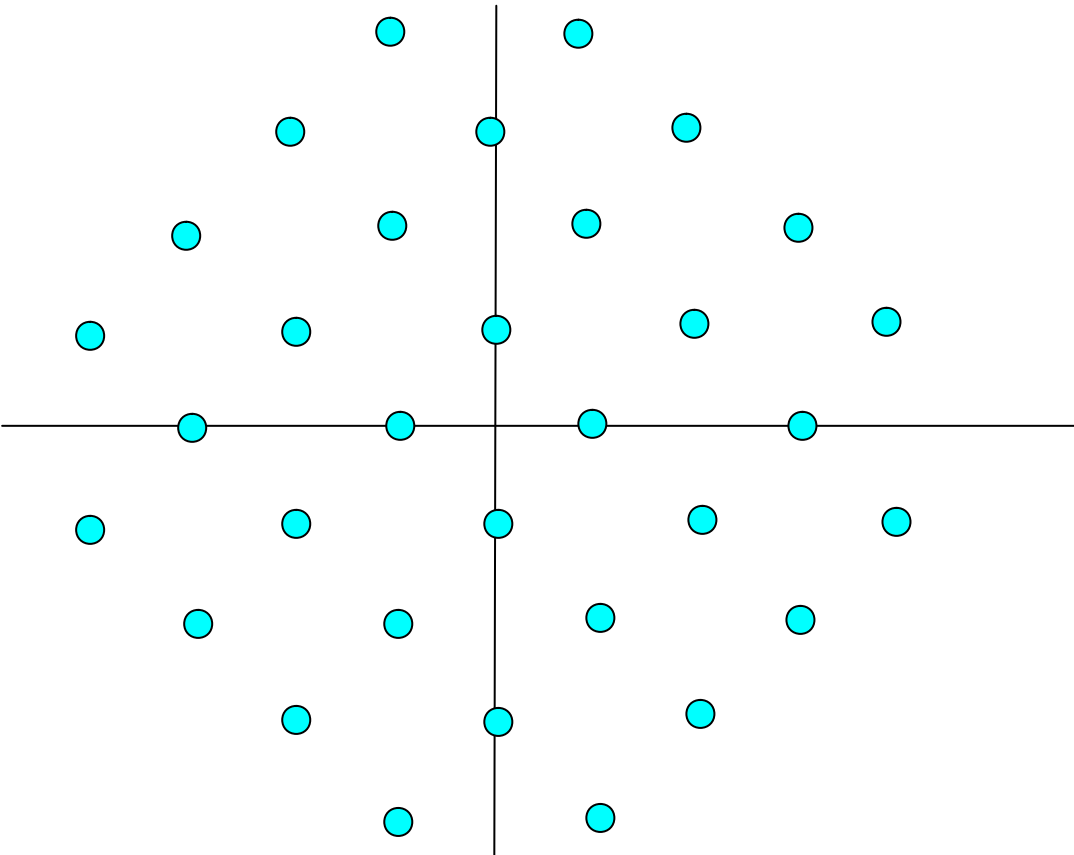
Diagramme spatial ou diagramme de Fresnel ou constellation



Exemple 2 : QAM ('Quadrature Amplitude Modulation) Modulation d'amplitude et de phase en quadrature

Modem V32 (valence 32 pour 4 bits de données et un bit de correction d'erreurs).

Notion de codage en treillis (TCM 'Treillis Coded Modulation')



## Tableau des normes de modems

Avis	Mode	Débit	Modul	Support	Trans
V21	Async	300b/s	Fr	RTC-LS2	Dup
V22	Syn/Async	600-1200b/s	Ph	RTC-LS2	Dup
V22bis	Syn/Async	1200-2400b/s	Am Q	RTC-LS2	Dup
V23	Syn/Async	1200-75b/s	Fr	RTC-LS2	Dup
V26	Syn	2400b/s	Ph	LS4	Dup
V27	Syn	4800b/s	Ph-D	LS4	Dup
V29	Syn	9600-4800b/s	Ph+A	LS4	Dup
V32	Syn/Async	9600-4800b/s	Am Q	RTC-LS2	Dup
V32bis	Syn/Async	14400-4800b/s	Am Q	RTC-LS2	Dup
V33	Syn	14400-12000b/s	Am Q	LS4	Dup
V34	Syn/Async	33600-2400b/s	Am Q	RTC-LS2	Dup

### Explications

- Syn : mode synchrone
- Async : mode asynchrone
- Am Q : modulation d'amplitude et de phase en quadrature.
- Ph : modulation de phase
- Fr : modulation de fréquence
- RTC : Réseau téléphonique commuté
- LS 2/4 : liaison spécialisée 2 ou 4 fils.
- Dup : mode bidirectionnel simultané

## Le modem V90 (56 Kb/s)

### Rappel

- Les modems traditionnels considèrent le RTC (Réseau Téléphonique Commuté) comme un réseau entièrement analogique.

- Le téléphone numérisé MIC avec un bruit de quantification des codecs de l'ordre de 36db offre un débit maximum théorique de l'ordre de 35 Kb/s

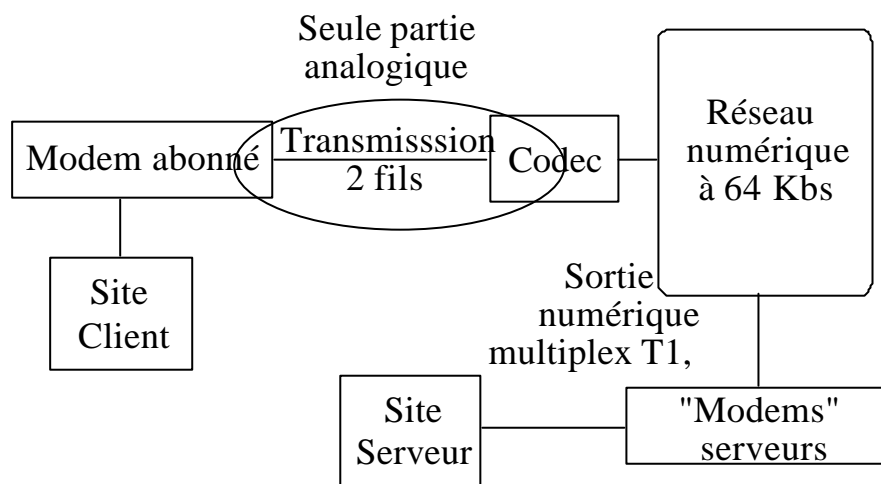
- D'ou la norme V34 constituant une limite pour cette approche avec un débit de:

33,6 Kb/s.

## Principes d'une nouvelle génération de modems sur RTC

a) En fait le réseau téléphonique est le plus souvent numérisé sur toute son étendue (sauf le rattachement abonné).

b) Les serveurs d'accès peuvent être directement rattachés au réseau numérique via des interfaces T1 (Etats-Unis) ou E1 (Europe).



=> La seule partie analogique est le rattachement usager sur lequel le rapport signal à bruit permet un débit supérieur à 33,6 kbs.

## La solution V90 (Rockwell , US Robotics)

- Le modem offre deux débits différents
  - 33 600 b/s sens client vers serveur  
(transmis en mode V34)
  - 56000 b/s sens serveur vers client  
(le plus intéressant à étudier)
- A 56 kb/s le coté serveur envoie des configurations numériques sur 8 bits vers le client.
- En fonction de la loi de quantification le codec génère des "symboles" analogiques (des niveaux de voltage)  
Rappel : deux lois de quantification existent A en Europe et  $\mu$  aux Etats-Unis.
- Le modem client interprète les signaux analogiques reçus pour reconstituer les octets transmis.

## Problème de mise en oeuvre

Le codec est conçu pour traiter des signaux de voix humaine: en particulier les niveaux de quantification les plus faibles sont privilégiés

D'ou un problème de discrimination dans le domaine des autres niveaux.

Utilisation d'une technique de codage tenant compte de cet aspect:

=> La technique de codage sur la partie analogique s'effectue avec perte de 8Kb/s ramenant le débit atteint à 56 Kb/s.

## Conclusion Modem 56 Kb/s

- Impossibilité (actuelle) de fonctionner à 56 Kb/s dans les deux sens:

-> peu grave pour des accès Internet car l'affichage est souvent privilégié par rapport à l'émission.

- Pour que le débit 56 Kb/s fonctionne il faut que le RTC soit entièrement numérique MIC et selon la même loi de quantification:

- . pas de section analogique

- . pas de conversion loi A loi m interne.

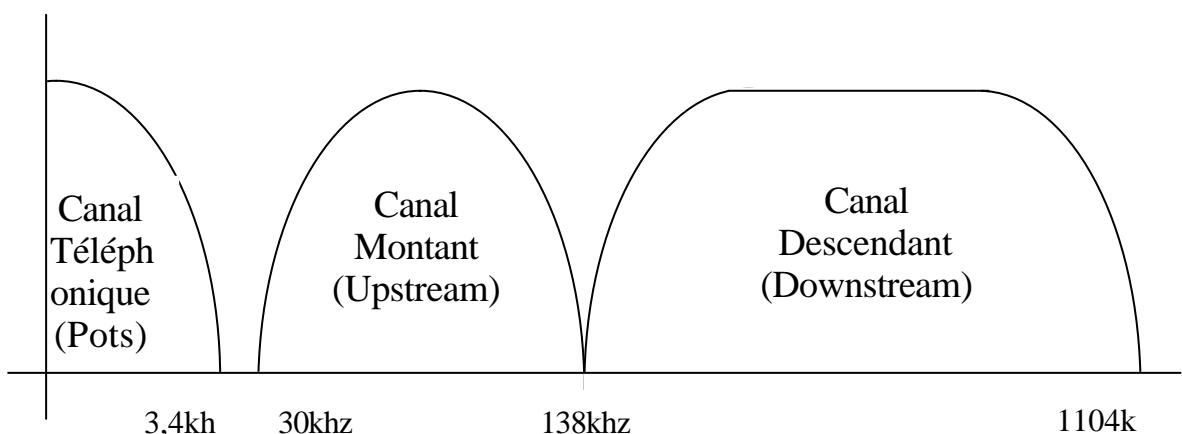
- . pas de section avec codage non MIC

(exemple pas de sections impliquant une conversion vers ADPCM)

- Lors de l'établissement de la communication les modems opposés doivent tester s'ils peuvent effectivement fonctionner à 56 Kb/s sinon le fonctionnement de repli est celui du V34 (ou éventuellement moins encore selon la qualité de la ligne).

## Le modem ADSL ('Asymmetric Digital Subscriber Line') ANSI T1.413, UIT G992.1

- Utilisation de la bande passante des paires téléphoniques de raccordement abonné autocom 1,1Mhz (sur moins de 3000 m, pas utilisable au delà de 6000 m).
  - Multiplexage fréquentiel de trois canaux (autre technique: annulation d'écho).
    - . 1) Canal téléphonique habituel (0 : 3,4khz) vers le réseau téléphonique
    - . 2) Canal numérique montant (upstream) abonné -> central (jusqu'à 1 Mb/s).
    - . 3) Canal num descendant (downstream) central -> abonné (jusqu'à 8 Mb/s).
- => Les débits sont asymétriques dépendant de la qualité et de l'offre (ex 128 / 512 Kb/s)**

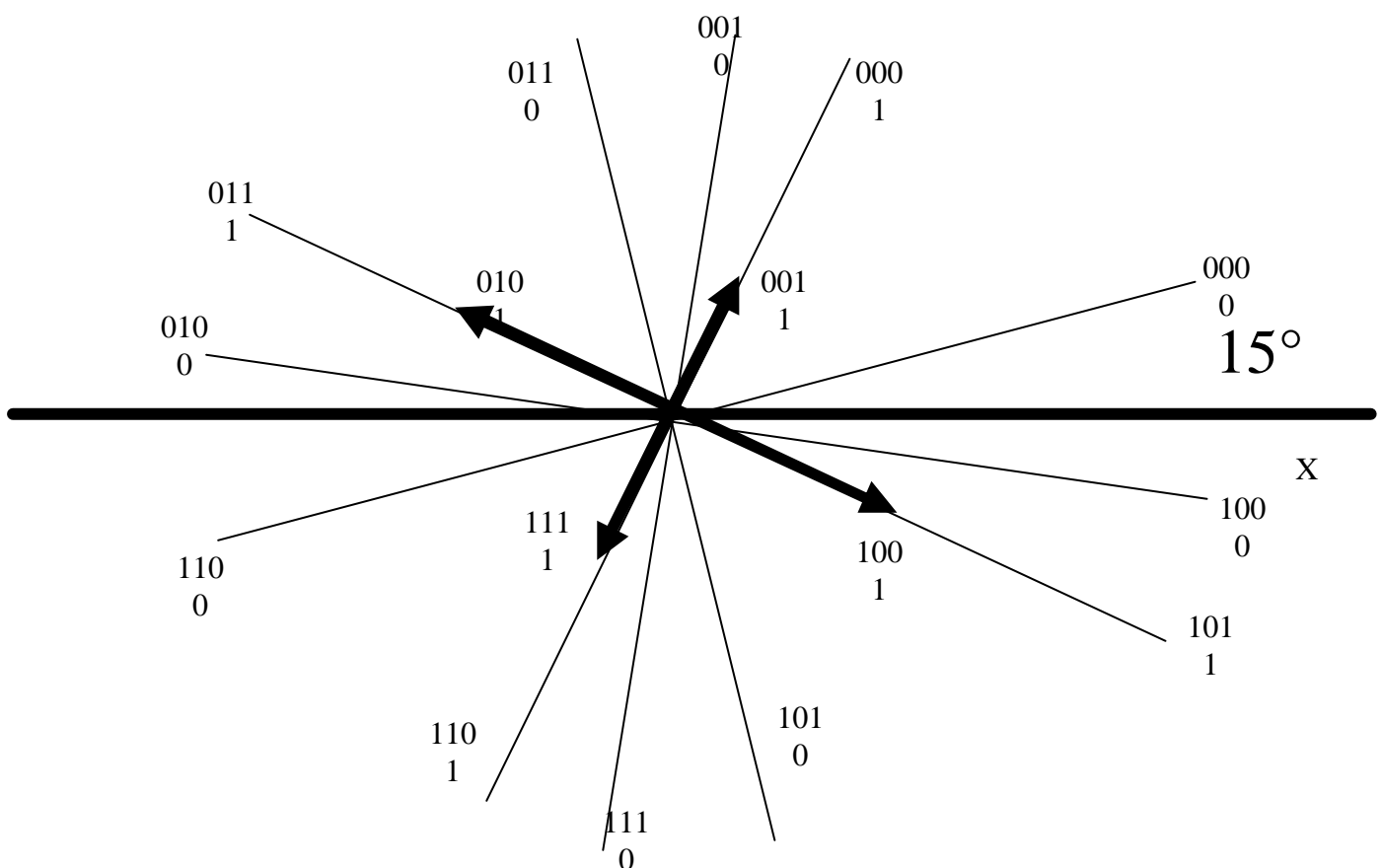




## Codage des voies numériques haut débit

### CAP 'Carrierless Amplitude Phase (modulation)'

- Modulation d'amplitude et de phase.
- Technique d'adaptation en débit à la qualité de la voie.
- Faible coût, faible latence, technique très classique (bien maîtrisée).

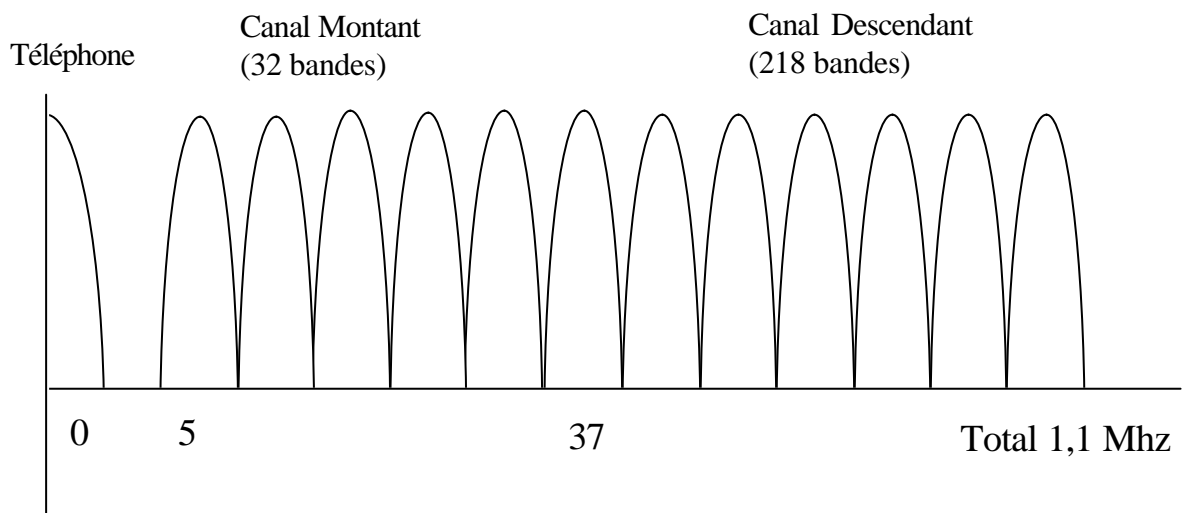


Modulation utilisée pour les 4 derniers bits : axe des x.

## DMT 'Discrete MultiTone'

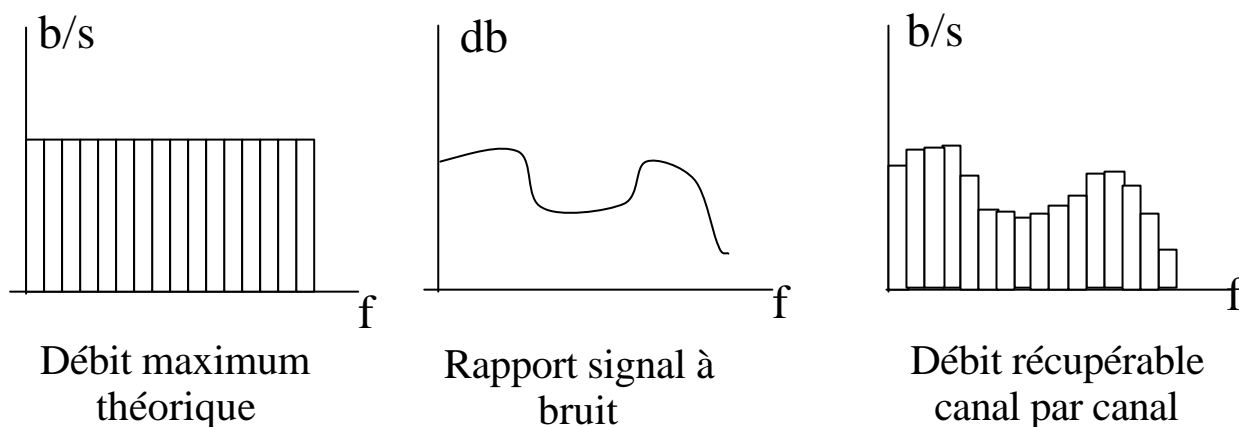
- **Technique plus fiable et plus sophistiquée, la plus élégante.**
- Division du spectre en 256 canaux de 4 KHz (4312,5 Hz), dépendant de l'opérateur.

Exemple de partage : 1 canal téléphonique, 5 non utilisés, 32 flux montant et 218 flux descendant gérées indépendamment en modulation d'amplitude et de phase.



## Technique de découverte adaptative du rapport signal à bruit (bande par bande).

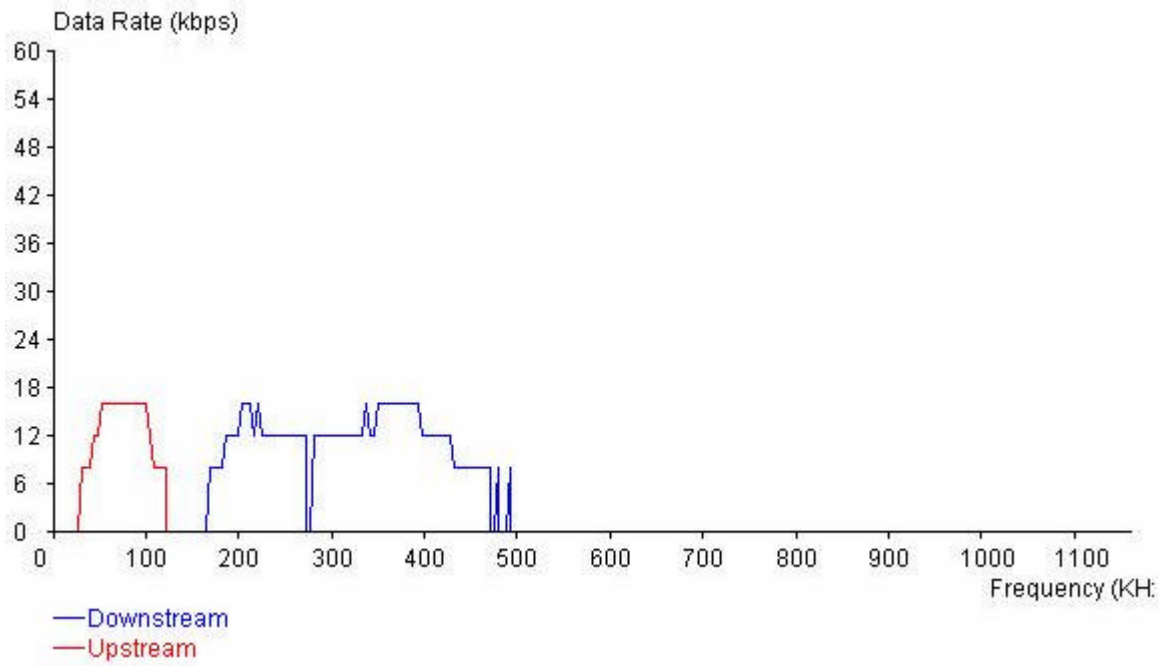
- La qualité de la transmission n'est pas la même pour deux abonnés, et dans chacun des 256 canaux (section de la paire torsadée, imperfections de la paire, longueur, ...)



- Débit théorique maximum par canal de 4kHz 60 Kb/s : soit pour le canal montant 1,5 Mb/s et pour le canal descendant 14,9 Mb/s
- En réalité quelques performances du canal descendant normalisées:

Débit	Section	Distance
1,5 Mb/s	0,5 mm	5,5 Km
6,1 Mb/s	0,4 mm	2,7 Km

# Un exemple de diagramme réel

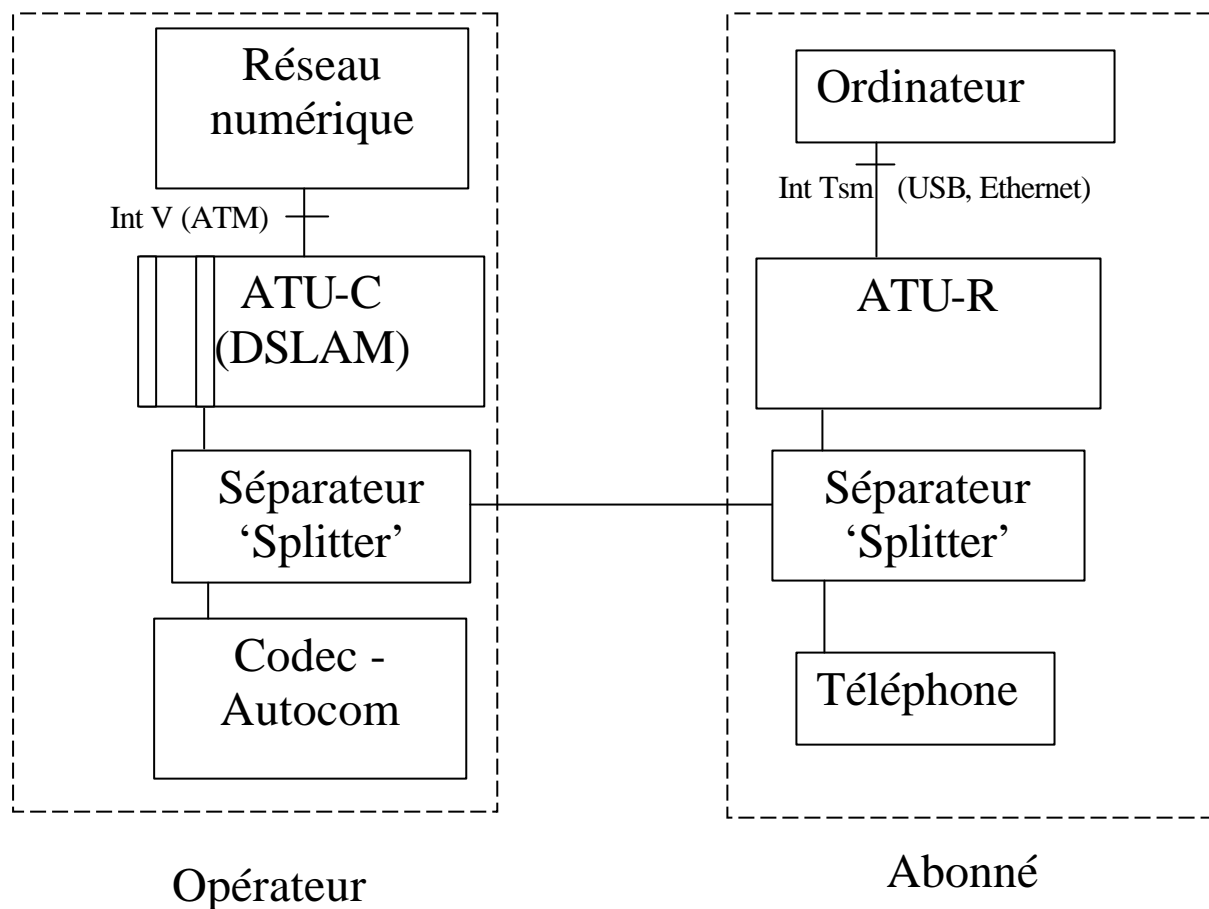




## **ADSL: un ensemble de techniques de transmission très complexe**

- **Tramage** (68 trames ADSL sont regroupées dans une super trame).
- **Synchronisation** des trames.
- **Embrouillage** des signaux.
- **Correction automatique** d'erreur par code correcteur d'erreurs dans les trames (FEC Forward Error Correcting Code Reed-Solomon).
- **Codage** (type QAM pour chaque canal, rapidité de modulation 4000 baud, définition de la constellation utilisée permettant jusqu'à 15 bits par intervalle ).

## ADSL: aspects architecturaux



- **ATU ADSL Transceiver Unit** (C Central, R Remote): l'essentiel du modem ADSL.
- **DSLAM Digital Subscriber Line Access Multiplexer** : multiplexeur d'accès ADSL chez l'opérateur.
- **Interface V** : nom générique ADSL de l'interface avec le réseau numérique de l'opérateur (exemple un réseau ATM +IP)
- **Interface Tsm** : nom générique de l'interface entre le modem ADSL et l'ordinateur usager (branchement sur port USB ou sur port ethernet).

# Les Supports de Transmission

## Standard EIA 568

"Commercial Building Télécommunications"  
Spécifie les caractéristiques des câblages

- Paires torsadées
- Fibres optiques
- Câbles coaxiaux (pour mémoire).

### Les paires torsadées

Deux conducteurs en cuivre, Isolés ,  
Enroulés de façon hélicoïdale autour de  
l'axe (l'enroulement permet de réduire les  
inductions électromagnétiques parasites de  
l'environnement).

#### **Utilisation courante**

Desserte des usagers du téléphone.

Réseaux locaux.



## **Paires torsadées blindées**

### **STP "Shielded Twisted Pairs"**

Fournies en câbles de 2 paires.

- Chaque paire est blindée
- L'ensemble est également blindé.

Vises des performances faibles en débit (boucle IBM à 16 Mb/s) Possibilités plus importantes -> 500 Mhz.

En fait nombreuses difficultés

- Coûteux à l'achat et à l'installation
- Assez encombrant
- Problèmes posés par les courants dans les blindages lorsque les appareils reliés sont à des potentiels différents.

Essentiellement recommandé par IBM dans l'installation de la boucle IBM avec le connecteur DB9 ou le connecteur IBM UDC "Universal Data Connector".

## Paires torsadées non blindées UTP "Unshielded Twisted Pairs"

Fournies en câbles de 4 paires.

Cinq catégories aux caractéristiques très normalisées:

- . impédance caractéristique
- . influence d'une paire sur l'autre en db
- . atténuation (en db).

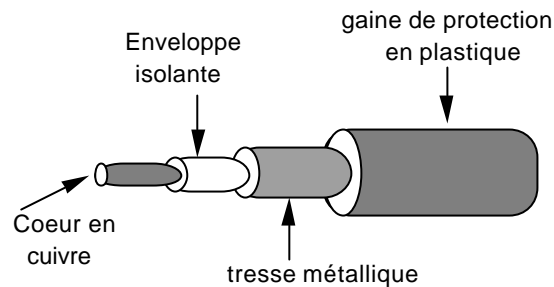
Catégorie	Bande Passante	Applications
UTP	Maximum	
1	Non définie	
2	Non définie	Téléphone V24/ RS232
3	16 Mhz	Ethernet 10BT 100VG, 100BT4 RNIS Déb base
4	20 Mhz	Boucle 16 Mb/s
5	100 Mhz	Ethernet 100BTX ATM 155 Mb/s

L'installation en catégorie 5 est la plus fréquente.

Elle doit être effectuée de façon conforme aux recommandations de pose pour tenir les débits au delà de 100 Mb/s

## Les câbles coaxiaux

Deux conducteurs concentriques: un **conducteur central** le coeur, un matériau **isolant** de forme cylindrique, une tresse concentrique **conductrice**:



- Meilleures caractéristiques électriques que les câbles à paires torsadés.

- Compromis entre la largeur de la bande passante et la protection contre les rayonnements parasites.

Centaines de MHz  $\leftrightarrow$  centaines de Mb/s

Exemple: 350 MHz  $\leftrightarrow$  150 Mb/s

Deux types de câbles coaxiaux dans les réseaux locaux:

- transmission en bande de base (50 ohms)
- transmission analogique (75 ohms).

**Actuellement en perte de vitesse au profit des paires ou des fibres.**

## Les fibres optiques

Bande passante très importante  
Affaiblissement très bas,  
Qualité de la transmission  
=> Les liaisons optiques sont très utilisées.

### Trois composants

- Conversion d'un signal électrique en signal optique => Une source de lumière,  
Diode électroluminescente  
(**LED** "Light Emitting Diode")  
ou Diode laser.
  
- Fibre optique qui joue le rôle d'un guide d'ondes lumineuses.
  
- Détecteur de lumière, photodiode de type **PIN** ("Positive Intrinsic Negative") ou à avalanche, qui restitue un signal électrique.

## Avantages des fibres

- Bande passante de l'ordre de 1 GHz/km (débit binaire très important).
- Faible affaiblissement de l'ordre de 1dB/km.
- Plusieurs dizaines de km entre amplificateurs.
- Insensible aux perturbations électromagnétiques
- Taux d'erreur très faible.
- Utilisation dans des environnements difficiles (variation de température...).
- Matière première bon marché.
- Légèreté et faible volume qui permettent la pose de tronçons longs avec plusieurs fibres.
- Possibilité de multiplexage en longueur d'onde (plusieurs "couleurs").

## Problèmes des fibres optiques

- Les raccordements (épissures optiques) restent délicats à réaliser sur le terrain et introduisent un affaiblissement d'environ 1dB.
  
- Les dérivations ne peuvent également s'effectuer qu'au prix d'une perte de puissance importante (limitation de l'utilisation de la fibre optique pour la diffusion).
  
- Les régénérateurs ou répéteurs, comportent un photodétecteur, un amplificateur électronique et une source lumineuse: l'amplification optique reste un sujet de recherche, ainsi que la commutation optique.

## **Les ondes en transmission à vue directe**

**Supports:** faisceaux hertziens , rayons infrarouges, rayons laser.

Une des solutions alternative à la pose d'une câble est l'installation d'un ensemble émetteur/récepteur sur des tours (les toits).

Les transmissions sont à faisceaux très directifs => Problèmes de météo.

### **Faisceaux hertziens**

Pour les communications longues distances, les faisceaux dirigés d'ondes radio à très haute fréquences constituent une alternative d'emploi des câbles coaxiaux.

Des antennes paraboliques sont installées au sommet de tours ou de pylônes et un faisceau hertzien est établi entre les deux antennes situées en vue directe à quelques dizaines de km.

## Les satellites de communication

Un satellite peut être considéré comme un **relais d'ondes** à très hautes fréquences (en fait plusieurs répéteurs).

### Répéteur

- **Écoute une fraction de la bande passante** de fréquences des signaux reçus par le satellite,

- **Détecte et amplifie les signaux** qu'il reconnaît

- **Réémet dans une autre bande** de fréquences.

Z Ceci évite toute **interférence** entre les divers canaux de transmission concernés.

### Satellite Géostationnaire

Un observateur le voit immobile par rapport à lui. A une altitude de 36 000 km au-dessus de l'Équateur, la période de rotation est de 24 heures. Elle est égale à celle de la terre.



## Compléments: satellites

- Écart minimum de position angulaire entre deux satellites inférieur à  $4^\circ$ .

=> évitement des interférences

Solution: n'utiliser qu'une partie des fréquences par satellite et avoir plusieurs satellites sur la même position angulaire.

- Les bandes de fréquences de 3,7 à 4, 2 GHz et 5,925 à 6,425 GHz sont attribuées aux satellites de transmission de données. Les bandes 12/14 et 20/30 GHz sont attribuées aux télécommunications.

- Un satellite utilise généralement un bande de fréquences de l'ordre de 500 MHz qui est partagée entre une douzaine de répéteurs, chacun d'eux n'utilisant qu'une sous-bande de 36 MHz.

- Chaque répéteur peut transmettre :
  - . un flot de données à 50 Mb/s,
  - . 800 canaux MIC à 64 Kb/s
  - . ou d'autres associations...

**Introduction au réseau  
téléphonique commuté  
(RTC)**

## Introduction à la transmission téléphonique numérique

La transmission numérique offre des performances supérieures à la transmission analogique:

- Faible taux d'erreur des liaisons numériques.

Les répéteurs numériques, régénérateurs de signaux, ... ne connaissent pas les inconvénients des amplificateurs utilisés par les supports analogiques.

- Les informations de type voix, images et données, représentées sous forme numérique peuvent facilement être multiplexées sur un même support de transmission.

Utilisation de cette technique pour le Réseau Numérique à Intégration de Services bande étroite:

Canaux de communication B à 64 Kb/s

## **Modulation par Impulsion et Codage MIC (PCM "Pulse code modulation")**

Le **codec** (codeur décodeur) convertit un signal analogique (voix parlée) en signal numérique et inversement.

### **Échantillonnage**

Bande passante visée  $H=4000$  Hz.

Nyquist  $2H = 8000$  échantillons/s

Un échantillon/125  $\mu$ s.

Échantillons sur 8 bits  $\Rightarrow$  Débit 64 Kb/s.

### **Quantification**

Établissement de la correspondance effective entre valeur du signal analogique et nombre sur 8 bits. Utilisation d'une loi (courbe semi-logarithmique) garantissant une précision relative constante.

En Europe Loi A. Aux USA Loi Mu

## Codage

Étape finale de représentation numérique des octets. Utilisation d'un transcodage.

**En Europe:** méthode ADI

"Alternate Digital Inversion"

1 bit sur 2 inversé 0000 -> 0101;

**Aux USA :** méthode ISC "Inverse Symmetrical Coding" 0011 -> 1100.

## **Autres procédés de modulation numérique de la voix**

Techniques de compression pour la quantification et le codage.

Ces techniques statistiques sont guidées par les propriétés de la voix humaine.

### **Modulation MIC différentielle (DPCM "Differential pulse code modulation")**

On transmet la différence entre la valeur présente et la valeur quantifiée précédente.

#### **Exemple 1: DPCM**

Si des écarts entre échantillons de  $\pm 16$  incréments, ou plus, sont très peu probables alors un codage sur 5 bits est suffisant.

Lorsque l'écart est supérieur à  $\pm 16$  incréments de quantification, il est nécessaire que l'encodage utilise plusieurs échantillons pour rétablir la situation.

#### **Exemple 2: Modulation DELTA**

On code sur un seul bit chaque échantillon en indiquant s'il est plus grand ou plus petit que le précédent de une unité.

Problèmes en cas de variations rapides.

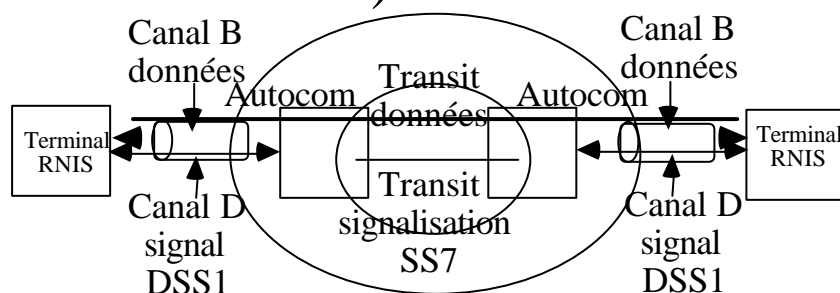
## Complément: Fonctions de signalisation du Réseau Téléphonique

Sur tout réseau téléphonique, nombreuses fonctions indispensables d'accès au réseau et d'administration (ex : l'utilisateur décroche, compose un numéro, le prestataire taxe ...) => Fonctions de **signalisation**.

Réseau téléphonique analogique: signalisation dans la bande => problèmes.

**Réseau téléphonique numérique:** un réseau numérique de signalisation (hors bande) utilise un protocole (système de signalisation) numéro 7 (SS7).

**Accès numérique usager (RNIS):** séparer la signalisation des données **des l'accès usager** (les signaux empruntent un canal spécial protocole DSS1).



# **Introduction aux réseaux d'infrastructures PDH-SDH**

## **Plan**

- Hiérarchies de multiplexage téléphonique
- Les problèmes posés par la transmission à haut débit en continu
- Introduction à la hiérarchie numérique plésiochrone (PDH)
- Introduction à la hiérarchie numérique synchrone (SDH)



## Les hiérarchies de multiplexage dans les réseaux téléphoniques

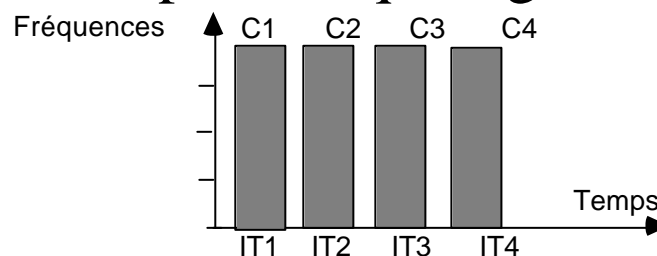
Pour optimiser les supports, le transport des circuits téléphoniques implique le multiplexage des circuits basse vitesse 64 Kb/s sur des voies haut débit.

Rappel: deux grandes classes de techniques

**Multiplexage par Répartition de Fréquence** (MRF ou FDM "Frequency Division Multiplexing") une bande de fréquence à chaque signal transporté.



**Multiplexage à Répartition dans le Temps** MRT ("TDM Time Division Multiplexing") une tranche de temps à chaque signal transporté.



## **A- Hiérarchie de multiplexage fréquentiel de circuits téléphoniques**

La transmission de la voix en mode analogique est basée sur le principe de la conversion des vibrations de l'air en signaux électriques d'environ 4000 Hz.

=> Bande passante nécessaire à la voix.

Dans un multiplexage fréquentiel primaire un groupe de 12 circuits de base est réuni: le signal composite résultant occupe 48 KHz (12 fois 4 KHz) à partir de 60 KHz.

Il s'agit du premier niveau de la hiérarchie de multiplexage fréquentiel du RTC ancien.

### **Différents niveaux de multiplexage**

<b>Structure</b>	<b>Bandes de fréquences</b>	<b>Capacité en circuits de 4 KHz</b>
Groupe primaire	60 à 108 KHz	12
Groupe secondaire	312 à 552 KHz	60
Groupe tertiaire	812 à 2044 KHz	300
Groupe quaternaire	8516 à 12 338 KHz	900

### **Hiérarchie ancienne abandonnée**

## **B- Hiérarchies de multiplexage temporel de circuits téléphoniques**

Un ensemble de normes de multiplexage et de transmission de canaux à 64 Kb/s défini entre 1970 et 1985.

### **Hiérarchie Numérique Plésiochrone PDH Plesiochronous Digital Hierarchy**

#### **Version européenne**

<b>Structure</b>	<b>Débit numérique</b>	<b>Capacité en circuits à 64 kb/s</b>	<b>Avis ITU-T</b>
Niveau-1	2 048 Kb/s	30	G732
Niveau-2	8 448 Kb/s	120 (4)	G.742
Niveau-3	34 368 Kb/s	480 (4)	G.751/1
Niveau-4	139 264 Kb/s	1920 (4)	G.751/2
Niveau-5	564992 Kb/s	7680 (4)	G.954/6

#### **Version américaine**

<b>Structure</b>	<b>Débit numérique</b>	<b>Capacité en circuits à 64 kb/s</b>	<b>Avis ITU-T</b>
Niveau-1	1 544 Kb/s	24	G733
Niveau-2	6312 Kb/s	96 (4)	G.743
Niveau-3	44 736 Kb/s	672 (7)	G752/1
Niveau-4	274 176 Kb/s	4032 (6)	

<b>Hiérarchie Numérique Synchrones</b> <b>SDH Synchronous Digital Hierarchy</b> <b>SONET Synchronous Optical Network</b>
--

De nouvelles techniques de multiplexage et de transmission définies à partir de 1984 aboutissent à une nouvelle hiérarchie.

### Version européenne

Nom	Débit numérique	Capacité en circuits à 64 kb/s	Avis ITU-T
STM-1	155,520 Mb/s	2016	
STM-4	622,080 Mb/s	8064	
STM-12	1866,24 Mb/s	24192	
STM-16	2488,32 Mb/s	33256	
STM-64	9510,912 Mb/s	129024	

### Version américaine (SONET)

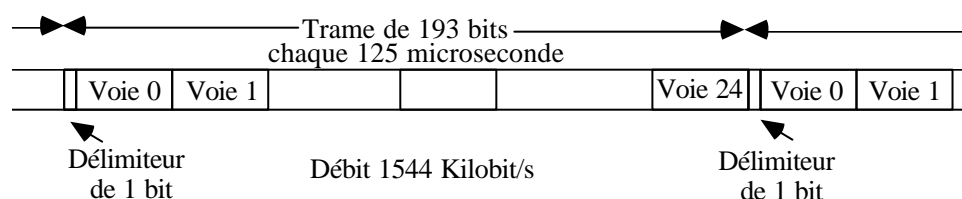
Nom	Débit numérique	Capacité en circuits à 64 kb/s	Norme
STS-1 / OC-1	51,84 Mb/s	672	
STS-3 / OC-3	155,520 Mb/s	2016	
STS-12/OC12	622,080 Mb/s	8064	
STS-48/OC48	1866,24 Mb/s	33256	
STS-192/OC1	9510,912 Mb/s	129024	

## Problèmes posés par la transmission à haut débit en continu

### Détermination de l'emplacement des informations significatives

Les infos multiplexées sont "tramées":  
assemblées dans des trames de taille fixe  
les trames sont émises en continu.

Exemple du multiplex T1 pour 24 voies  
téléphoniques MIC (États-Unis)



=> Il faut pouvoir retrouver les trames en réception en présence de multiples aléas de transmission (bruits, désynchronisations ...).

Notion de "**verrouillage de trame**":

Une trame est dite "verrouillée" pour le récepteur s'il est correctement synchronisé sur son début.

# Différences des fréquences d'horloge

## Différences de fabrication des horloges

Les informations sont générées avec des horloges qui ne génèrent de manière parfaites des fréquences identiques.

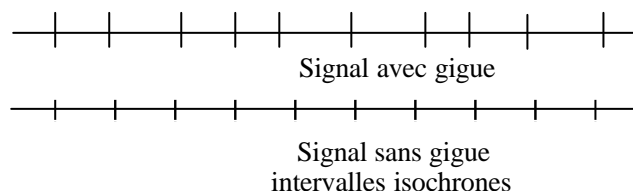
Cependant les horloges employées doivent être suffisamment précises.

Exemple: l'horloge pour générer le multiplex PDH à 2,048 Mb/s doit-être à plus ou moins  $5 \cdot 10^{-5}$  ou encore  $\pm 50$  ppm (parties par million)

## Gigue ("Jitter")

La fréquence d'un signal peut présenter des variations instantanées (autour d'une fréquence moyenne).

Par exemple lorsque des retards différents sont introduits dans le traitement des signaux.



## **Dérive d'horloges ("Wander")**

La fréquence d'une horloge dépend de la température et peut donc subir des dérives lentes (à l'échelle de plusieurs heures ou même plus) autour d'une valeur moyenne.

La dérive d'une horloge doit-être bornée.

### **Déphasages**

Les signaux issus de lignes différentes présentent des déphasages.

### **Débits**

Les différents appareils intervenant dans une chaîne de transmission ne traitent pas exactement les informations à la même vitesse.

### Régénération de signal en entrée

- Élimination de la gigue par utilisation de mémoires tampons.
- Lissage des fréquences d'entrée par utilisation d'oscillateurs à boucle de fréquence asservie ou à verrouillage de phase (PLL "Phase Lock Loop").

### Génération de signaux en sortie régulés

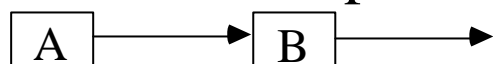
- A partir d'un tampon alimenté irrégulièrement, asservissement d'une horloge d'extraction dont le rythme est le plus régulier possible.

Ces technique ne permettent pas de traiter tous les cas possibles en particulier celui de la fabrication d'un multiplex à haut débit à partir de plusieurs affluents présentant des caractéristiques hétérogènes.



## Les techniques de justification: Positive, Nulle ou Négative

Deux sites reliés par une voie haut débit:



- Les horloges de A et B sont **identiques**.

**Pas de justification**

On réémet le **même nombre** de bits.

- L'horloge de A est plus rapide que celle de B => Justification **négative**

Pour un nombre de bits donné arrivant en B de A on doit en renvoyer **plus**.

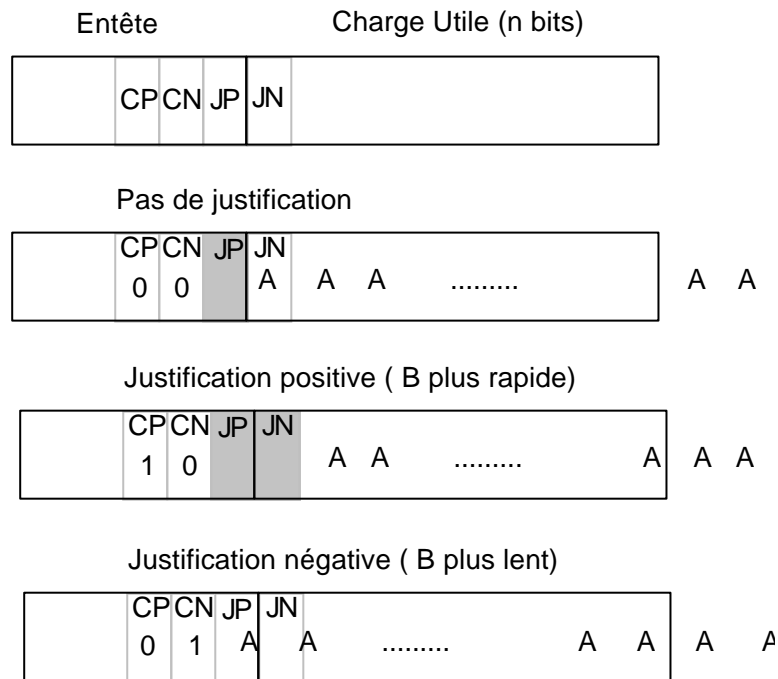
- L'horloge de A est plus lente que celle de B => Justification **positive**

Pour un nombre de bits donné arrivant en B de A on doit en renvoyer **moins**.

Techniques de justification:  
appliquées aux réseaux PDH et SDH

# Fonctionnement de la justification

Présentation de la **justification par bits**.  
 CP: bit de contrôle de justification positive  
 CN: bit de contrôle de justification négative  
 JP: bit de justification positive  
 JN: bit de justification négative

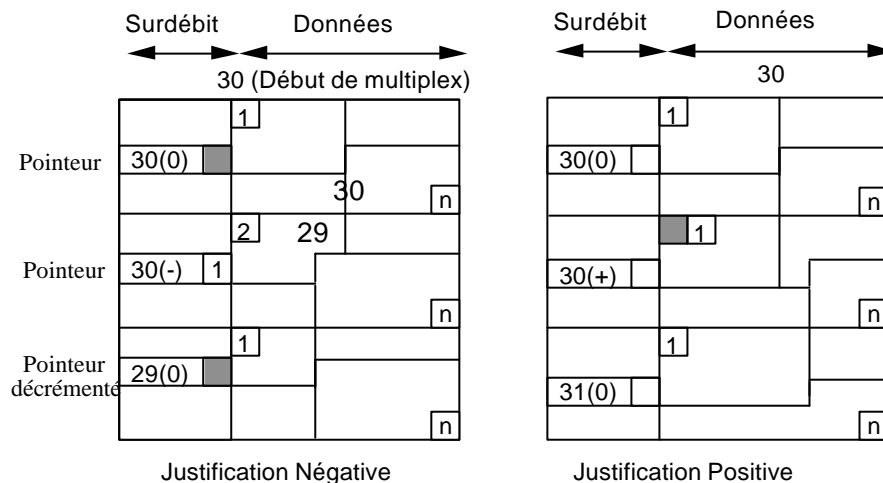


Si des trames successives sont soumises à la des justifications successives les délimitations entre les zones transportées glissent à l'intérieur des conteneurs de charge utile  
 => Utilisation de **pointeurs**.

## Utilisation de pointeurs

Un pointeur, participant au **surdébit**, est associé à la charge utile de la trame. Sa valeur ne change pas tant que la position relative de cette charge utile dans la trame ne varie pas.

Le début d'informations utile peut avancer ou être retardé par rapport à la valeur initiale de sorte que la charge utile peut “flotter” dans l'espace alloué à l'intérieur de la trame.



Horloge locale plus lente

Horloge locale plus rapide

Remarque: Présentation comme en SDH, la justification porte sur des octets et les trames ont une représentation matricielle).

## Compléments gestion des pointeurs

Si l'horloge locale a une fréquence plus faible que celle du noeud d'origine du signal transporté:

- On associe au pointeur une **indication de justification négative**.

- Un octet de donnée est retranché vers le surdébit.

- Dans la trame suivante la valeur du pointeur est diminuée de 1.

### En cas contraire

- Un octet de bourrage est inséré dans la charge utile.

- On associe au pointeur une **indication de justification positive**.

- Dans la trame suivante le pointeur est augmenté de 1.

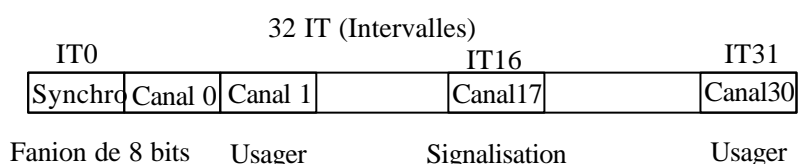
# Introduction aux réseaux PDH

## Plésiochronous Digital Hierarchy

Hiérarchie numérique **plésiochrone**.

Les horloges sont indépendantes les unes des autres. La resynchronisation s'opère entre sites voisins.

Exemple: le multiplex E1 à 2 048 Kb/s  
G704 Européen de niveau 1 (TN1)



Les trames comprennent:

- Un **surdébit**: nécessaire au maintien et à la récupération du synchronisme: l'IT0 acheminé dans le premier octet entête.

Utilisé en version multiframe il offre des possibilités de transmission d'informations multiples et complexes.

- Un **débit utile**: ici on a 30 voies téléphoniques MIC et une voie IT16 permettant l'acheminement de la signalisation pour les 30 voies.

## Utilisation du surdébit (IT0)

En regroupant 16 trames (multiframe) on dispose de 128 bits chaque 2 millisecondes utilisés pour de nombreuses fonctions:

- **Une zone de verrouillage de trame** pour définir le début des trames (la synchronisation trame).
- Une zone pour la **justification bit**.
- Une zone pour des **signaux d'alarme** (en cas de mauvais fonctionnement).
- Une zone de **contrôle de qualité** (au moyen d'un **CRC-4**).
- Des zones libres d'usage spécifiques (par exemple à caractère national).
- etc

## Conclusion PDH

Encore largement utilisée dans une base installée importante.

Les développements se sont arrêtés dans les années 1980 avec l'arrivée de SDH.

# Introduction aux réseaux SDH

## Synchronous Digital Hierarchy

### SONET Synchronous Optical Network

Manque de flexibilité de la hiérarchie plésiochrone: pour accéder à une donnée dans un multiplex il faut démultiplexer tous les niveaux supérieurs:

=> Développement à partir de 1980 d'une nouvelle technologie: **la SDH.**

Tous les noeuds d'un réseau utilisant la hiérarchie numérique synchrone sont **synchronisés** avec des horloges de référence d'une précision spécifiée à **10<sup>-11</sup>**.

-> On a encore des dérives sensibles (utilisation à très haut débit).

L'originalité de la technologie SDH repose sur la technique **des pointeurs**

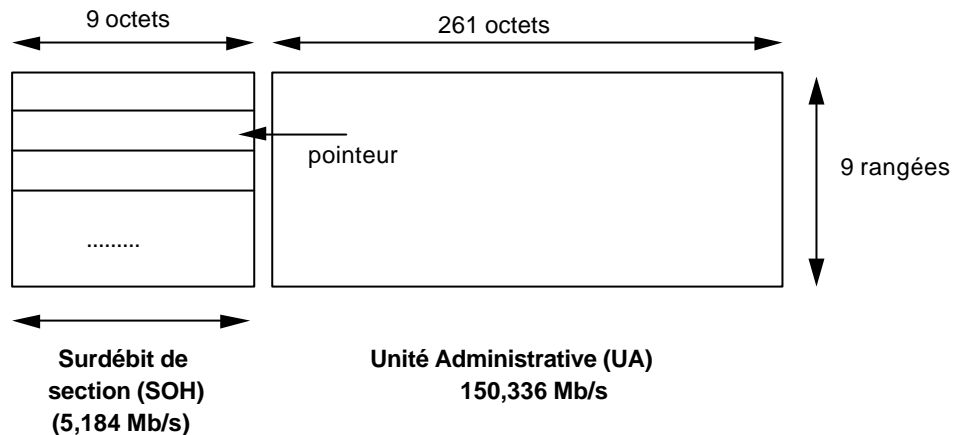
- ils permettent de compenser les problèmes de variation de fréquence du signal
- mais aussi d'accéder directement aux données à l'intérieur des trames par des mécanismes d'indexation simples ...

## Multiplexage primaire STM-1 SDH

STM-1 "Synchronous Transport Module 1"

Trame de base constituée de 2430 octets émis toutes les 125  $\mu$ s, soit 155,520 Mb/s.

Organisée en 9 rangées de 270 octets.



- Dans une trame STM-1, les informations sont placées dans des conteneurs qui peuvent être vus comme une structure hiérarchisée de groupage. Le conteneur + son surdébit forment un conteneur virtuel (VC<sub>n</sub>).

STM1 -> Un conteneur VC4 ou plusieurs conteneurs plus petits: VC-1 (1,544 à 2, 048 Mb/s), VC-2 (pas encore normalisé) , VC-3 (34,368 à 44,736 Mb/s).



## Utilisation actuelle de SDH

Le niveau STM1, conteneur VC-4 est le plus important.

Transport de cellules ATM, multiplexs plésiochrones ou d'autres types de signaux (signaux vidéo, cellules DQDB,...).

La hiérarchie SDH est encore en développement.

- Débits de plus en plus importants.
- Introduction d'une architecture de réseau complète utilisant SDH au niveau physique avec des répartiteurs, des brasseurs, des multiplexeurs.

**Introduction au Réseau Numérique à  
Intégration de Services RNIS  
et à l'interface S**

**ISDN "Integrated Digital Service Network"**

## Objectifs

Construire par étapes à partir du réseau téléphonique un réseau multi-services incluant voix, données, images.

- Numérisation du réseau téléphonique
- Modernisation de la signalisation
- RNIS-BE bande étroite (64 kb/s->2 Mb/s)  
"Narrowband ISDN" N-ISDN
- RNIS-LB large bande (2Mb/s -> ...)  
"Broadband ISDN" B-ISDN

Le RNIS BE englobe différents concepts:

- Une connexité complète à 64 Kb/s par la numérisation du lien usager-réseau.
- La possibilité d'utiliser un système de signalisation puissant, rapide et disponible.
- Des interfaces normalisées offrant des services intégrés à partir d'un même point.

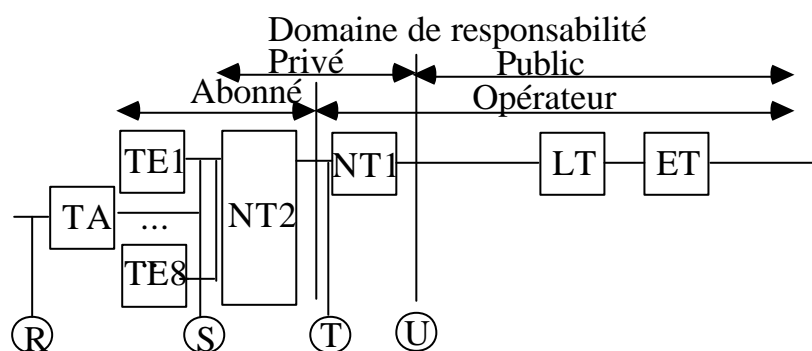
## RNIS Interface à l'accès de base

Le RNIS emprunte la ligne de raccordement usager-réseau (interface U) utilisée aujourd'hui pour le téléphone analogique.

De part et d'autre de ce raccordement on trouve des fonctions de gestion de signaux :

- **Côté abonné** : L'équipement terminal de ligne **NT1** ("Network Termination 1").

- **Côté central** : l'équipement terminal de ligne **LT** ("Line Termination") qui connecte le commutateur **ET** ("Exchange Termination") à la ligne de raccordement.



L'équipement terminal de ligne (NT1) permet de connecter au travers d'une interface au point de référence **T** un terminal RNIS appelé **TE** ("Terminal Equipement" au sens ETTD), ou un bus d'abonné sur lequel 8 terminaux peuvent se raccorder.

Un équipement dit **NT2** ("Network Termination 2") peut être connecté au NT1. Ce peut être un autocommutateur privé (**PABX**) ou une régie d'abonné jouant reliant les terminaux de l'abonné et le RNIS.

**R** désigne une connexion entre un terminal non RNIS et l'adaptateur de terminaux **TA** aux normes RNIS.

**U** désigne la connexion de l'équipement terminal de ligne et de l'autocom le plus proche.

**S** désigne une connexion entre un terminal RNIS et le dispositif de raccordement **NT2**.

**T** désigne la connexion entre le dispositif de raccordement **NT1** et un appareil d'abonné ou le **NT2**.

## **Multiplexages et débits sur les interfaces de références S et T**

Chaque interface de référence usager-réseau représente un multiplexage de canaux A,B,D,E, H dont les débits sont différents.

Par ailleurs des informations de service (verrouillage de trame, contrôle d'accès, ...) circulent dans des bandes baptisées Y.

- A - canal 4 kHz de téléphone analogique.
- B - canal 64 kB/s MIC pour voix et données.
- C - canal 8 ou 16 kB/s numérique.
- D - canal 16 ou 64 kB/s de signalisation.
- E - canal 64 kB/s signalisation interne RNIS
- H - canal 384, 1536, 1920 kB/s numérique.

### **Interface d'accès de base So ou To**

2 canaux B 64 + 1 canal D 16

### **Interface d'accès au débit primaire S2 T2**

30 canaux B 64 + 1 canal D 64

## Interface d'accès de base So ou To Spécifications Physiques et de débits

- Interface 4 fils formée d'une paire émission et une paire réception avec la possibilité de téléalimenter les terminaux à concurrence de 400mW (énergie suffisante pour un téléphone).

- Le connecteur normalisé (ISO 8877) dispose de 8 broches, quatre sont réellement utilisées, deux par sens de transmission.

- Le câble utilisé entre les terminaux et l'équipement de terminaison NT1 doit comporter des conducteurs ayant un diamètre d'au moins 0,6 mm.

- Débit utile 144 Kb/s sur trois canaux :

- . deux B de 64 Kb/s connexions numériques de bout en bout.

- . un D de 16 Kb/s signalisation

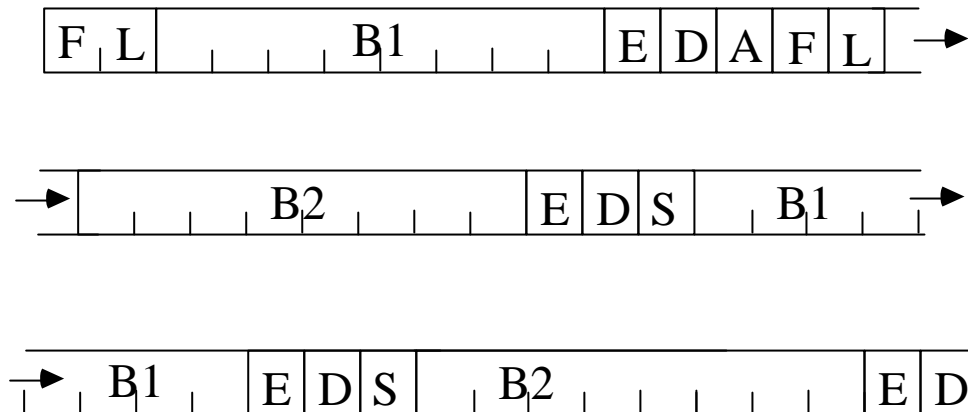
(**DSS-1** Digital Signalling System 1).

- Le débit réel est de 192 Kb/s.

La différence de 48 Kb/s est utilisée pour la gestion du bus passif (résolution de contention et activation des terminaux) ainsi qu'au multiplexage dans le temps des trois canaux (**2B+D**).

## Structure de trame So

- Trame de 48 bits chaque 250 microseconde



- 36 bits de données usagers 16 B1, 16 B2, 4D soit 144 kb/s.

- 12 bits de service

Bits F : synchronisation trame

Bit L : rétablissement de l'équilibre électrique

Bits E : écho des bits D

Bit A : activation désactivation d'équipement

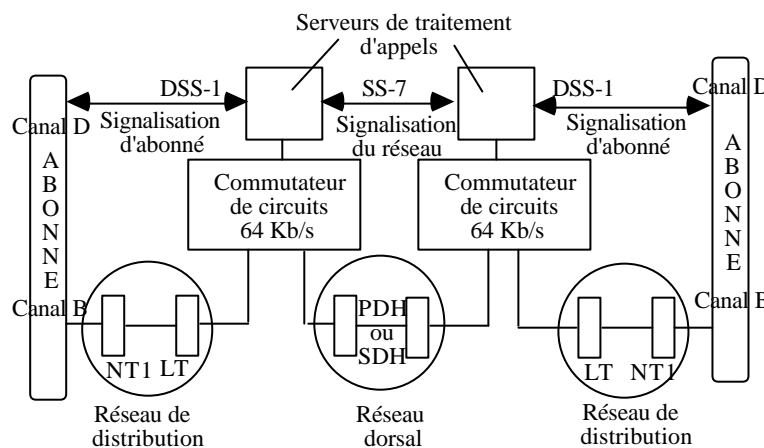
Bits S : spare non utilisé



# Signalisation

La signalisation d'abonné (DSS-1), se prolonge dans le réseau selon **le système de signalisation CCITT n°7** encore appelée **signalisation sémaphore**.

Celui-ci fonctionne sans connexion et utilise le réseau spécifique appelé réseau sémaphore ou réseau de signalisation.



DSS-1 est découpé en deux niveaux fonctionnels pour échanger entre l'utilisateur et le réseau sur le canal D des données en mode paquet.

- un niveau liaison LAPD de type HDLC.
- un niveau réseau protocole D offrant un très grand nombre d'options de signalisation.

## Services offerts par le RNIS

- Les services support;
- Les téléservices
- Les compléments de service

### Services supports

Deux services correspondent au débit et à la qualité de transport garantis par le réseau pour l'acheminement d'une communication de bout en bout.

- Le service de commutation de circuits (CC) sur canal B avec transmission numérique non transparente (NT).

CCBNT ne garantit pas l'intégrité de la séquence de bits au travers du réseau : des techniques de compression peuvent être utilisées sur ce circuit pour en réduire le débit tout en respectant la nature du signal analogique. Pour la téléphonie.

- Le service de commutation de circuit (BB) sur canal B en transparence (T).

CCBT garantit l'intégrité des données. Il est adapté aux transmissions de données.

## Téléservices

Le numéro d'annuaire caractérise un accès sur lequel plusieurs terminaux de types différents (téléphone numérique, télécopieur, ordinateur...).

L'appel entrant est alors présenté par diffusion à tous les terminaux. Chacun d'eux vérifie sa compatibilité avec le profil du terminal appelant relayé par le réseau, et seul le terminal compatible répond

Si plusieurs terminaux sont compatibles, des compléments d'adresse peuvent être envoyés par l'appelant.

Les attributs de téléservice sont :

- **Téléphonie 3 kHz**, équivalente au service téléphonique actuel et compatible avec lui.
- **Télécopie** groupe 2, 3 ou 4
- **Télétext** en mode caractère et en mode mixte (caractère et télécopie)
- **Vidéotex pour** connecter un minitel,
- **Vidéotéléphonie**  
(signal vidéo à balayage lent)

## Les compléments de services

Ils sont disponibles en tant que compléments de services support ou téléservices (exemple reroutage si l'utilisateur demandé est occupé).

Ces compléments peuvent être demandés à la souscription ou invoqués appel par appel.

- **Sélection directe à l'arrivée** : associé aux autocommutateurs privés (PBX)

- **Sous-adressage** : permet de transférer un complément d'adresse en dehors du plan de numérotage et donc de sélectionner un terminal particulier.

- **Signalisation d'utilisateur à utilisateur** : associée à une communication, elle fournit une liaison virtuelle sur canal D entre appelant et appelé en plus de la connexion de circuits sur le canal B.

- **Identification du demandeur** : permet à l'appelé de connaître l'adresse de l'appelant pour un filtrage par le demandeur

## RNIS Conclusion

Le développement du RNIS bande étroite est lent pour de nombreuses raisons techniques et économiques:

- Existence d'investissements considérables sur le réseau téléphonique classique et coûts très importants de passage au RNIS.

- Possibilités d'adapter le réseau téléphonique pour lui faire supporter la plupart des applications visées par le RNIS.

- Les solutions techniques RNIS déjà anciennes ne répondent pas aux besoins vers les hauts débits qui s'affichent de plus en plus (réseaux locaux, images animées).

Autres solutions réseaux (niveau 2-3)

Relais de trames

ATM relais de cellules

Éventuellement Internet

- Problèmes de tarification: le RNIS peut-il être offert à un tarif attractif pour les usagers et rentable pour les prestataires de service.

# **LE NIVEAU RÉSEAU**

## **I Problèmes généraux du niveau réseau**

## **II Protocoles industriels**

### **- IP**

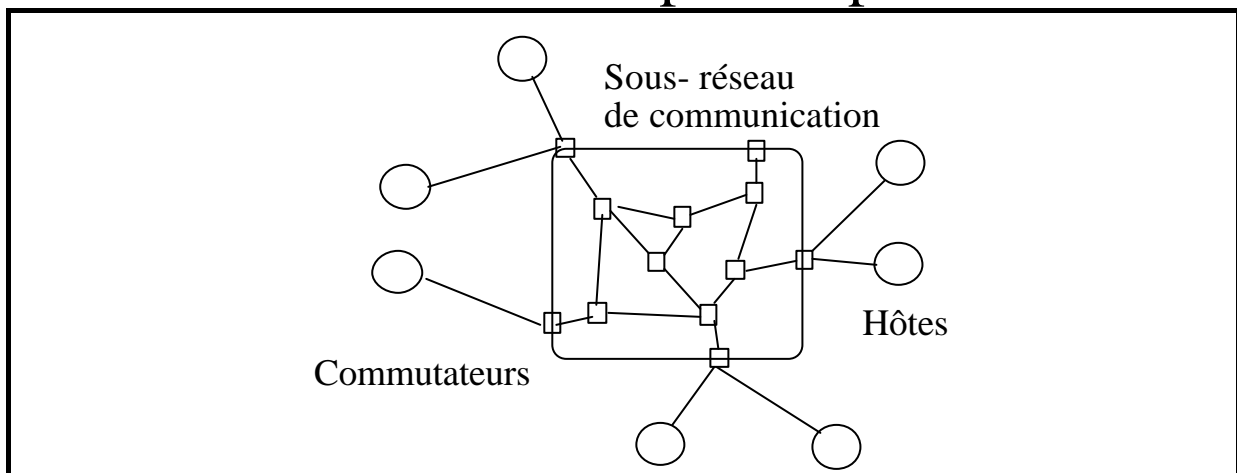
# Problèmes généraux du niveau réseau

## Introduction

### Fonctions principales du niveau 3

Le niveau réseau assure l'acheminement des paquets d'un **hôte émetteur** vers un **hôte destinataire** (tous deux connectés à un réseau de communication).

- Ceci implique en général de traverser plusieurs commutateurs et donc d'assurer dans chaque commutateur la fonction de **routage**.
- Entre les commutateurs on utilise les services de liaison point à point.



## Fonctions réalisées à l'intérieur de la couche réseau

Selon les options de conception d'une couche réseau une partie des fonctions suivantes est réalisée:

- ***Routage.***
- ***Segmentation.***
- ***Multiplexage***  
(des connexions de réseau sur des connexions de liaison).
- ***Correction des erreurs***  
(d'hôte à hôte).
- ***Contrôle de flux*** (d'hôte à hôte).
- ***Satisfaction de contraintes de qualité de service***
- ***Respect de l'ordre d'émission***  
(livraison en séquence d'hôte à hôte).
- ***Gestion des connexions.***
- ***Diffusion***



# Organisation architecturale

## Positionnement de la fonction routage

### - *Modèle OSI, INTERNET*

La couche 3 est dédiée au routage.

### - *Couches liaison et réseau intégrées*

Dans certaines choix architecturaux pour des raisons de performance les couches liaison et réseau sont réunies en une seule couche  
Exemples: Arpanet version 1 (protocole IMP-IMP), relais de trames, ATM.

### - *Réseaux locaux*

En raison des possibilités du protocole d'accès au médium un routage vers un ensemble de destinataires est réalisé par le niveau liaison (commutation de niveau 2).

## Terminologie

**Commutation:**      => Relayage

Utilisation de techniques plutôt matérielles.

**Routage :**              => Mise à jour des tables

Utilisation de techniques plutôt logicielles.

## Services demandés par la couche réseau à la couche liaison

### Au minimum

- *Échange d'unités de données* en point à point entre sites voisins (reliés par une liaison spécialisée ou un réseau local).

### Fonction complémentaires importantes

- *Contrôle d'erreurs*

De transmission en point à point  
Signalement des erreurs irrécupérables.

- *Contrôle de flux* entre sites voisins

- *Livraison en séquence*

Des unités de données

- *Multiplexage de flots*

Support de plusieurs protocoles réseaux  
sur la même liaison.

## Services offerts par le niveau réseau au niveau transport

### Au minimum

- *Échange d'unités de données*  
entre sites reliés par un réseau

. Ceci suppose une couche réseau avec un adressage uniforme et un routage.

=>

Le niveau transport doit alors être indépendant de la nature et des possibilités des sous-réseaux effectivement traversés ...

### Fonction complémentaires

- **Selon les choix de conception** du réseau les services offerts sont plus ou moins riches.
- Le travail de la couche transport pourra être **plus ou moins complexe.**

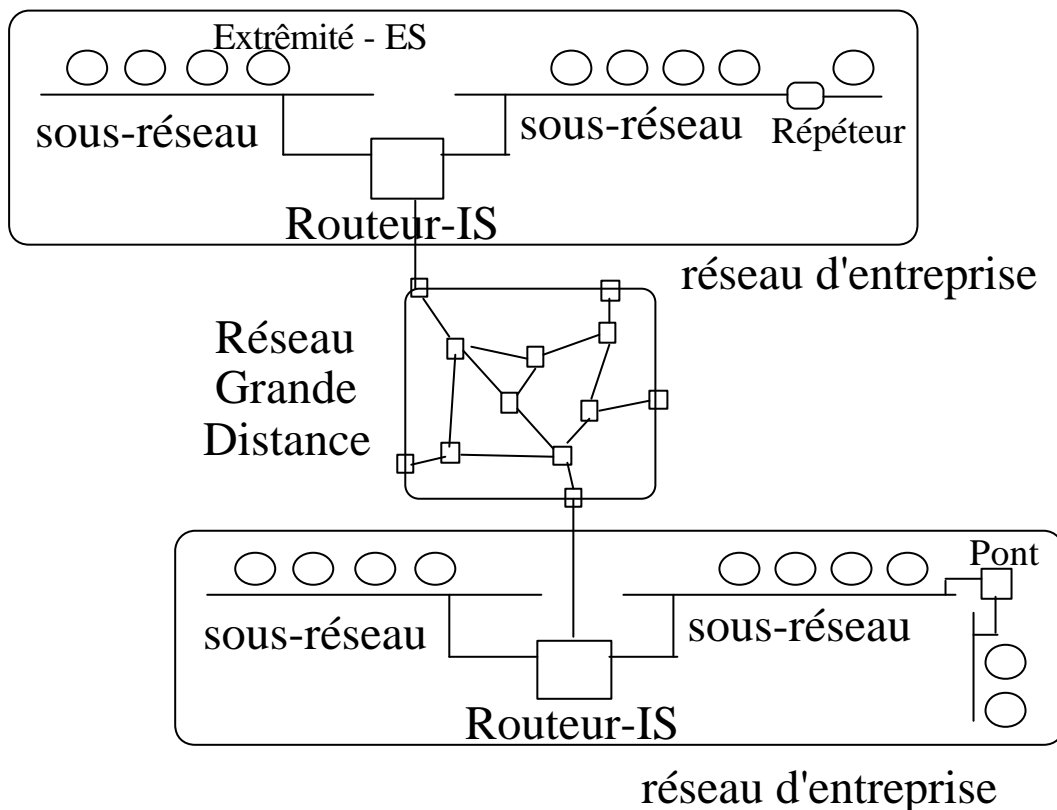
## Le problème d'interconnexion de réseaux

- Permettre à des **hôtes** (ES "End System") rattachés à **des réseaux différents** (même type ou type différent) de communiquer.

- En utilisant des commutateurs particuliers permettant de passer d'un réseau à l'autre.

. IS "Intermediate System"

. Routeurs ("routers")



- **Adressage universel** commun aux réseaux.

- **Conversion des formats** utilisés.

# Plan du cours

## Chapitre 1

### Problèmes généraux de réalisation du niveau réseau

1. Choix de conception
2. Adressage
3. Routage
4. Contrôle de congestion

## Chapitre 2

### Exemples industriels

1. IP: Internet Protocol
2. ATM: Asynchronous Transfer Mode

# I Choix de conception d'un niveau réseau

Possibilité de retenir de nombreuses options de conception d'une couche réseau:

- **Contrôle d'erreurs**  
de transmission d'hôte à hôte.
- **Reprise sur panne du réseau**  
pour assurer une bonne sûreté réseau.
- **Contrôle de flux** (d'hôte à hôte).
- **Respect de l'ordre local** d'émission  
lors de la délivrance.
- **Mode connecté ou non connecté**

L'expérience a conduit à privilégier trois ensembles cohérents d'options:

## *1 Les réseaux à datagrammes*

## *2 Les réseaux à circuits virtuels*

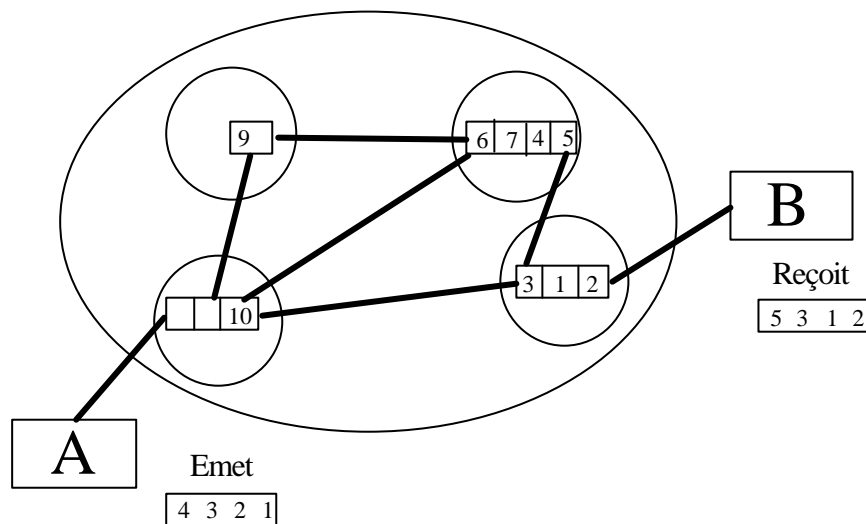
### *2.1 A bas débit*

### *2.2 A haut débit*

# Réseaux à datagrammes

## Modèle analogue à celui de la poste

- Les datagrammes sont des paquets **roulés indépendamment** les uns des autres.
- Chaque datagramme comporte dans son entête **l'adresse du destinataire**.



**Choix coûteux du point de vue du routage**  
=> Chaque datagramme subit un routage.

**Choix permettant plus d'optimisation**  
=> On peut tenir compte des informations les plus récentes pour réaliser le routage.

## Options des réseaux à datagrammes

Un ensemble de choix simplificateurs (sauf routage) pour une couche "**mince**".

Exemple : Internet Protocol IP

### - **Pas de livraison en séquence:**

. Non respect de l'ordre local d'émission du site A lors de la délivrance en B

. Les messages ne se suivent pas.

### - **Pas de contrôle de flux** d'hôte à hôte

. Difficulté de rétroaction du destinataire sur l'émetteur.

### - **Pas de contrôle d'erreurs au niveau 3**

. Pas d'hypothèses fortes sur le niveau liaison qui peut ne pas corriger les erreurs de transmission, ni respecter l'ordre local, ni contrôler le flux entre les commutateurs.

### - **Difficultés de gestion des ressources**

(réservation des tampons, de la bande passante)

. Problèmes de congestion.

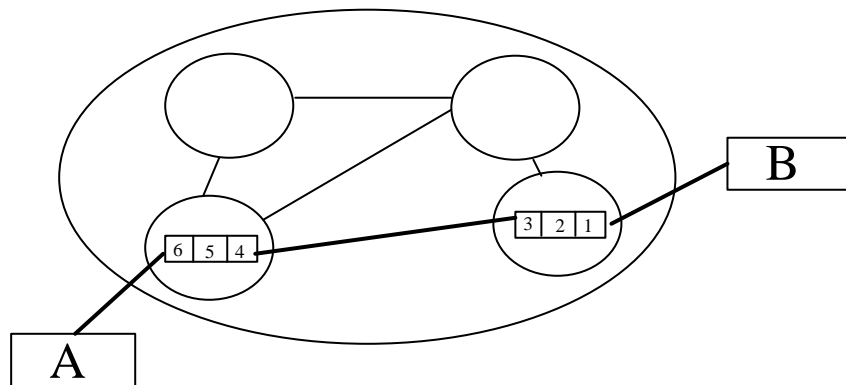
**=> Les problèmes non résolus sont donc reportés au niveau transport**



## Réseaux à circuits virtuels

### Modèle analogue à celui du téléphone

- Un circuit virtuel est un **chemin fixe** établi entre deux hôtes.
- Le chemin est initialisé par un premier paquet: "**paquet d'appel**" qui est routé à partir de l'émetteur en fonction de l'adresse du destinataire.
- Les données empruntent **toujours ce chemin** (tant qu'il n'y a pas d'incident)  
=> **Simplification des opérations de routage** pour chaque paquet.



- En cas d'incident, il faut **reconstituer** un nouveau circuit virtuel.

## Options des réseaux à circuits virtuels Bas débits

### Exemple : **X25** niveau paquet

Un ensemble d'options assez complètes pour  
une couche réseau consistante.

- **Livraison en séquence** (d'hôte à hôte)  
Respect de l'ordre local d'émission du site A lors de la délivrance en B (les paquets se suivent sur le chemin).
  - **Contrôle de flux** (d'hôte à hôte)  
Par rétroaction sur le chemin.
  - **Allocation préalable de ressources**  
Assure le maintien d'une qualité de service constante pour un circuit.
  - **Hypothèses sur le niveau liaison:**
    - . Détection et correction des erreurs,
    - . Respect de l'ordre local,
    - . Contrôle de flux entre les commutateurs

=> Très peu de problèmes de transmission subsistent sauf les pannes du réseau.
- => Le niveau transport peut-être plus simple.

# Options des réseaux à circuits virtuels Hauts débits

## Deux exemples

### *Relais de trames* *FR, "Frame Relay"*

Un allègement des choix X25 (réseaux publics de transmission de données) afin de permettre la commutation efficace de trafics de plus hauts débits.

### *ATM "Asynchronous Transfer Mode* *TTA Technique temporelle asynchrone*

- Réseau numérique à intégration de service large bande.
- Transmission de données multimédia image, voix, données.

Satisfaction des contraintes de qualité de service pour ces trois types de données.

### *MPLS "Multi Label Switching"*

## **Choix techniques des réseaux à circuits virtuels hauts débits**

- . Utilisation de **circuits virtuels**  
Avantage: la rapidité de commutation.
  
- . Communication en mode **connecté**  
Associé au mode circuits virtuels
  
- . Livraison en séquence  
Paquets à la suite selon un circuit virtuel
  
- . **Sans contrôle d'erreur**  
Médium fibre optique: taux d'erreur faible  
Contrôle d'erreur inadéquat pour les trafics  
isochrones (sons, images).
  
- . **Sans contrôle de flux**  
Besoins différents pour les trafics  
isochrones (sons, images).
  
- . Couche supplémentaire **d'adaptation**  
Spécifique du type de données échangées.  
Garantit un certain type de service.  
**Qualité de service**

## Niveau Réseau en mode connecté ou sans connexion

### Mode connecté

Les échanges dans une communication en mode connecté ne peuvent avoir lieu qu'entre deux événements ouverture et fermeture de connexion.

Exemples: X25 niveau paquet, ATM

=> **Délimitation dans le temps des échanges.**

=> **Désignation d'une connexion.**

La connexion est identifiée par une référence unique. Les messages circulent selon une connexion.

=> **Définition d'une qualité de service associée à la connexion.**

Caractérisée par des paramètres quantitatifs

- . débit,
- . taux d'erreur,
- . facteur d'anticipation

...

ou des paramètres qualitatifs

- . profil de service utilisable

## Mode non Connecté

Les échanges dans une communication en mode non connecté peuvent prendre place à tout moment.

Exemple: IP Internet Protocol

=> **Pas de délimitation temporelle** des échanges.

=> **Désignation explicite** des extrémités communicantes dans tous les messages.

=> **Unités de données auto suffisantes.**

Tous les paramètres de qualité de service à respecter doivent être disponibles dans le paquet

=> Limitation des possibilités.

## **Relations entre réseaux à circuits virtuels et communications connectées**

- La **désignation** des connexions ou des CV.
- La gestion de **qualité de service**.
- La correction des **erreurs** de transmission.
- La livraison en **séquence** d'hôte à hôte.
- Le contrôle de **flux** d'hôte à hôte, la QOS.
- **L'allocation des ressources**.

## **Relations entre mode datagrammes et communications non connectées**

- **Pas** de livraison en **séquence** d'hôte à hôte.
- **Pas** de **contrôle de flux** d'hôte à hôte.
- **Pas** de **gestion des ressources**.
- **Pas** de **contrôle d'erreurs**.

## Remarque / Terminologie

Pas d'équivalence complète entre mode connecté et circuits virtuels ou mode non connecté et datagrammes.

- Possibilité de construire un protocole en mode connecté sur un réseau à datagrammes (puisque'on peut communiquer).

- Utilisation dans un réseau à circuits virtuels comme X25 du routage du paquet d'appel pour transmettre des datagrammes (option "fast select").

**Il est impropre d'utiliser les équivalences**

**Datagramme = Mode non connecté**

**Circuits virtuels = Mode connecté**



## **2. LE PROBLÈME D'ADRESSAGE AU NIVEAU RÉSEAU**

# Les problèmes d'adressage (1)

## Adressage / désignation

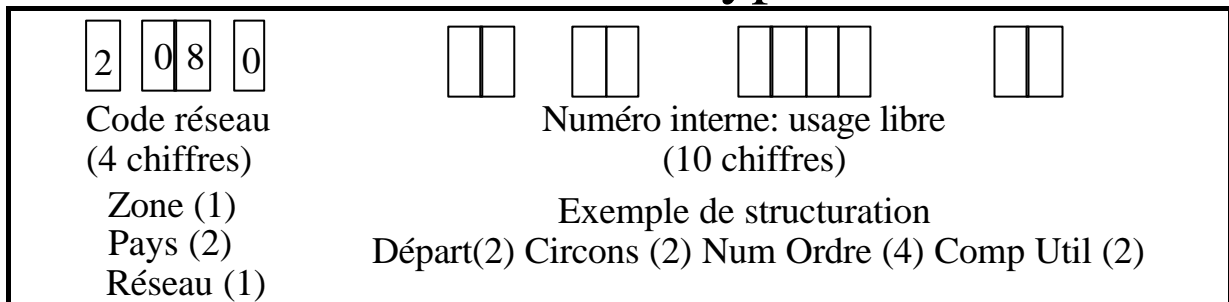
("adressing", "naming") au niveau 3 (réseau):

L'ensemble des techniques associées aux adresses et aux noms des sites dans un réseau

### 1 Politique de **structuration**

des différentes zones d'une adresse.

Exemple d'adresse X121 des réseaux publics de transmission de données type TRANSPAC.



2 Définition des **autorités** administratives compétentes dans l'attribution des adresses.

Exemples "d'autorités": Organismes internationaux divers (INTERNIC "Internet Information Center", ...).

Opérateurs de télécoms, fournisseurs d'accès ("carriers", "providers") réseaux.

## Les problèmes d'adressage (2)

3 Politique de **stockage et d'accès** aux différents moyens de désignation.

Gestion d'annuaires

Gestion des tables de routage.

4 Politique d'utilisation des adresses dans le cadre des opérations de **routage**.

### -> **Cas des datagrammes**

Dans chaque paquet figurent:

- l'adresse de l'émetteur
- l'adresse du destinataire.

### -> **Cas des circuits virtuels**

Dans le paquet appel figurent:

- l'adresse de l'émetteur
- l'adresse du destinataire.

Dans chaque paquet figure ensuite:

- l'identifiant du circuit virtuel.

## Nom ou Adresse / Physique ou logique / Fixe ou mobile

Différents problèmes enchevêtrés dans la désignation des sites:

**Nommer:** Identifier de façon unique

**Adresser:** Retrouver dans un réseau

**Physique:** Identification associée à des aspects matériels invariants (numéro de série)

**Logique:** Identification d'un site selon une chaîne de caractères quelconque.

**Fixe:** Qui ne peut suivre le déplacement de l'appareil connecté (numéro de téléphone d'une ligne du réseau commuté attachée à un autocommutateur).

**Mobile:** Qui permet de retrouver indépendamment de la localisation (numéro d'un téléphone portable non rattaché à une cellule).

## Adressage global / Adressage local

Pour aller à un endroit on peut:

- Donner l'adresse de l'endroit à atteindre.

=> Notion d'**adresse "globale"**

Mr X, appartement y, z rue de .....

- Définir le chemin à emprunter:

=> Notion d'**adresse "locale"**

Tourner à gauche puis 2 ième à droite ....

### **Adressage global**

Un identifiant unique du site destinataire est acheminé dans chaque message pour déterminer son routage et sa délivrance.

Exemple: adresse IP, adresse ATM.

### **Adressage local**

Un chemin est défini par la suite des décisions à prendre lors de la traversée de chaque commutateur.

Exemple: identification des circuits virtuels en X25, ATM ....

# **STRUCTURATION DES ADRESSES GLOBALES**

# Adressage plat

## Définition

Les adresses sont définies dans une zone de n bits sans règles de structuration particulière.

## Avantages

*-Adapté à de petits réseaux.*

=> Peu de problèmes d'administration.

*-Les adresses sont courtes .*

=> Minimisation de l'encombrement dans les messages.

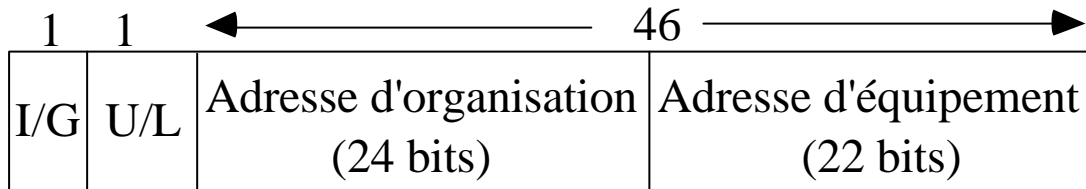
## Inconvénients

*-N'offrent pas de moyen lié à la structure des adresses pour retrouver un correspondant => Inadapté aux grands réseaux.*

## Exemple d'adressage plat

### Adressage IEEE 8.02

Adresse standard des réseaux locaux  
(format rare sur 16 bits  
surtout employé sur 48 bits).



I : Adresse individuelle (=0)

G : Adresse de groupe (=1)

L : Adresse Locale (=0)

U : Adresse Universelle (=1)  
Attribuée sur demande aux IEEE



## Compléments: adressage IEEE 8.02

- Aucune préoccupation dans la définition de la structure des adresses ne permet de l'utiliser pour retrouver un hôte d'un réseau.

- Par contre volonté de gérer la distribution des adresses en permettant aux utilisateurs de créer:

### **. des adresses à la demande**

Les adresses commençant par 00 sont déterminées par les administrateurs de systèmes.

### **. des adresses uniques**

(au niveau mondial).

Les adresses commençant par 01 sont uniques.

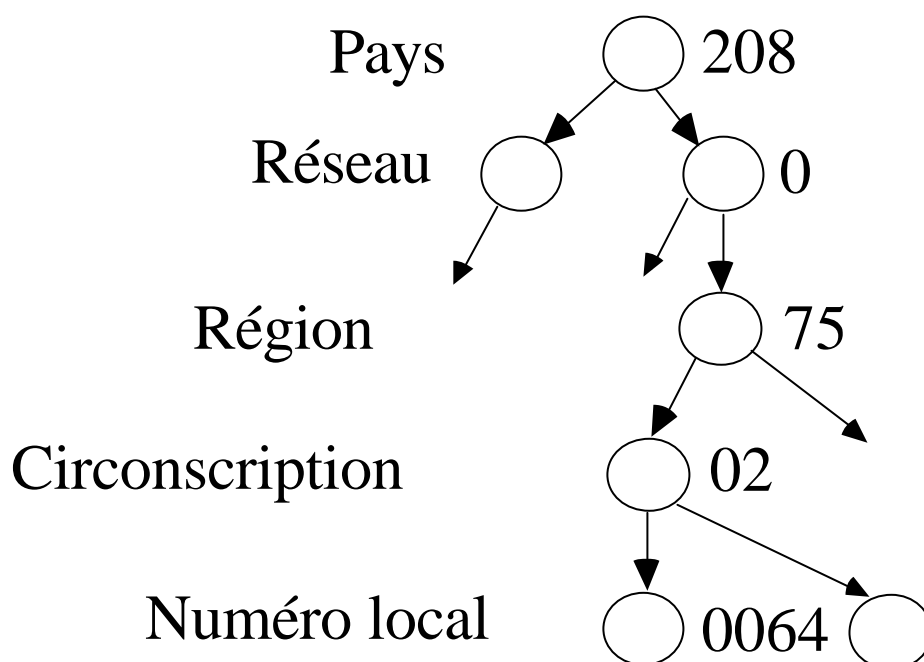
Pour alléger la distribution d'adresses (initialement assurée par XEROX puis par les IEEE) définition d'une zone identification d'organisation ("Organizationally Unique Identifier" OUI ) qui identifie des entreprises habilitées à délivrer des numéros uniques.

## Adressage hiérarchique

- Les adresses sont définies avec des règles de **structuration** par champs correspondant à des regroupements :

- . géographiques (pays, région, ...)
- . logique (domaine, sous domaine,...).

- A la manière des adresses téléphoniques ou des noms de fichiers.



## Avantages de l'adressage hiérarchisé

### - *Adapté aux grands réseaux*

Possibilité de gérer de grands espaces.

L'administration des adresses peut se faire de façon locale à chaque unité de découpage.

### - *Adapté à la localisation*

Les adresses permettent de localiser le destinataire dans des grands réseaux.

Le routage peut-être hiérarchisé.

## Inconvénients

### - *Encombrement*

Le découpage opéré peut-être plus ou moins efficace mais généralement les adresses sont volumineuses.

En fait dans un adressage hiérarchique on a une grande perte de capacités d'adressage (gaspillage d'adresses).

### - *Changement de localisation*

Un site qui change de localisation doit changer de nom.

Pour supporter la mobilité il faut mettre en place un mécanisme complexe de redirection.

## Exemples d'adressages hiérarchiques

Tous les grands systèmes d'adressage niveau3

### Exemple 1: Adressage E164

- Définition de l'essentiel des adressages réseaux des grands opérateurs.

. RTC: Réseau Téléphonique Commuté

POTS " Plain Old Telephone System"

. RNIS : Réseau Numérique à Intégration de Service.

. ATM : Réseaux ATM publics.

- Basée sur un système de 15 chiffres décimaux maximum codés en BCD (Binary Coded Decimal) deux chiffres par octets.

- Forme générale d'organisation d'adresse.

Code pays	Code destination nationale	Numéro d'utilisateur
Country code	National Destination Code	Subscriber Number

### Exemple 2: Adressage Internet

- Deux normes d'adressage correspondent aux deux versions successives majeures:

IPV4 Adresse sur 32 bits.

IPV6 Adresse sur 128 bits.

- Traité en cours plus loin.

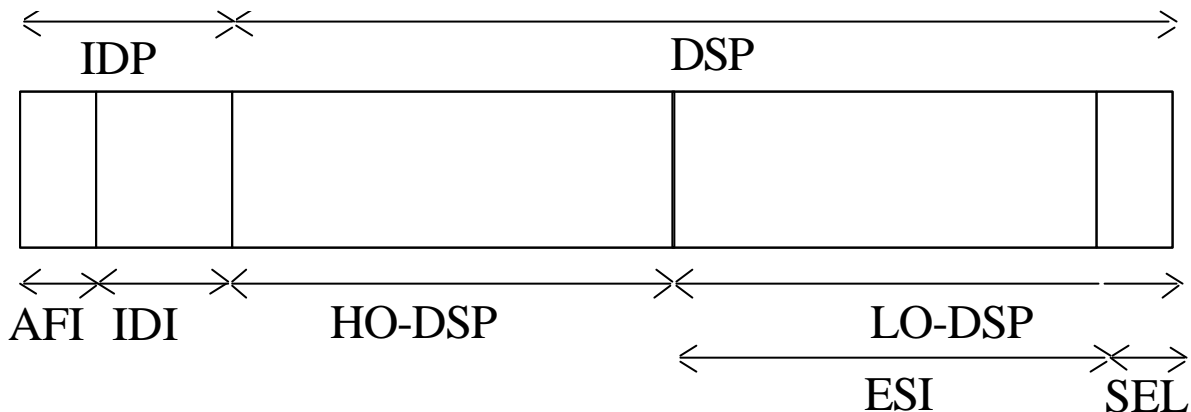


**Exemple 4 (adressage hiérarchique)**  
**Adressage normalisé OSI**  
**Norme ISO 8348 ou ITU-T X213**

Format variable en deux parties pour des adresses de NSAP.

**("Network Service Access point")**

Taille maximum 20 octets.



***IDP: "Initial Domain Part"***

Spécifie le format de l'adresse (AFI) puis l'autorité responsable de l'attribution de cette adresse (IDI).

***DSP: "Domain Specific Part"***

Spécifie plus particulièrement un site.

Séparé en trois parties: "High Order Bits" (pour le routage), une partie identificateur de site ("End System Identifier) plus un sélecteur d'application (SEL) ("Low Order Bits").

## Précisions relatives à l'adressage OSI

### *AFI: "Authority and Format Identifier"*

- Sur deux chiffres décimaux le type de l'adresse définie dans la suite.

Exemple: Réseaux publics de transmission de données (code 36)

Réseau téléphonique, RNIS, ...

### *IDI : "Initial Domain Identifier"*

Première partie de l'adresse définissant le domaine dans lequel l'adresse est définie:

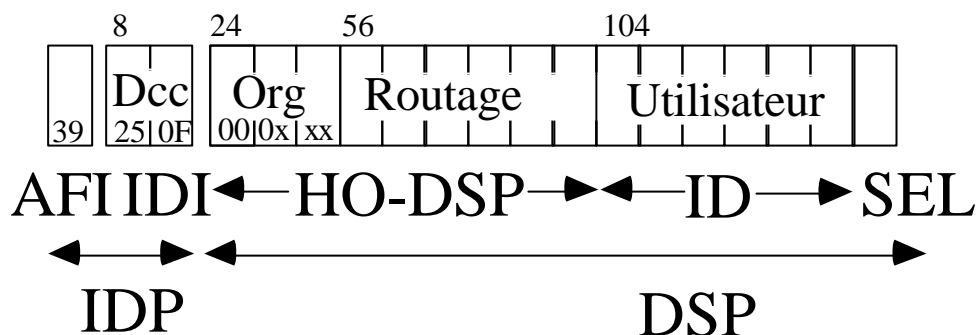
Exemple : Pour une adresse téléphonique l'indicatif du pays.

### *DSP : "Domain Specific Part"*

Le numéro proprement dit à l'intérieur du domaine.

## Un exemple d'adresses OSI Les adresses ATM forum au format DCC ("Data Country Code")

L'un des types d'adresses ATM pour des  
réseaux ATM privés  
(recommandé par l'ATM forum).



### Usage du champ IDP

- AFI "Authority and Format Identifier"  
Ici la valeur effective est 39.

- IDI en fait le DCC "Data Country Code"

Code sur deux octets des différents pays  
(norme OSI 3166)

Exemple: France 250 + bourrage F (250F)



## Usage du champ DSP

- Les différents pays sont responsables de l'administration des adresses DCC.

- En France c'est l'AFNOR qui gère les adresses.

=> L'AFNOR attribue des numéros (ici notés x) d'organisation (ou d'entreprise) sur 6 chiffres complétés à gauche par des 0.

Exemple d'organisation: le réseau de la recherche haut débit français RENATER II

- Le reste de l'adresse est sous la responsabilité de l'organisation .

. La fin de la zone HO-DSP permet en général de préciser le plan d'adressage et de routage (cas du réseau RENATER II).

. La zone ID identifie à l'intérieur d'une entreprise un site utilisateur

Elle peut-être encore être découpée (en domaines et sous domaines).

**ADRESSES LOCALES DANS LES  
RÉSEAUX A CIRCUITS VIRTUELS**

## **Problème d'identification d'un circuit virtuel**

Comment désigner un chemin pour aller d'un point à un autre dans un réseau?

### **Nom unique global**

. Une possibilité (adresse appelant, adresse appelé, numéro CV) => Très encombrant.

Exemple: 30 digits pour des adresses X121 plus un numéro (si plusieurs CV).

. N'apporte pas de solution particulière pour le routage.

### **Nom contextuel local**

#### ***Gestion locale de noms de petite taille***

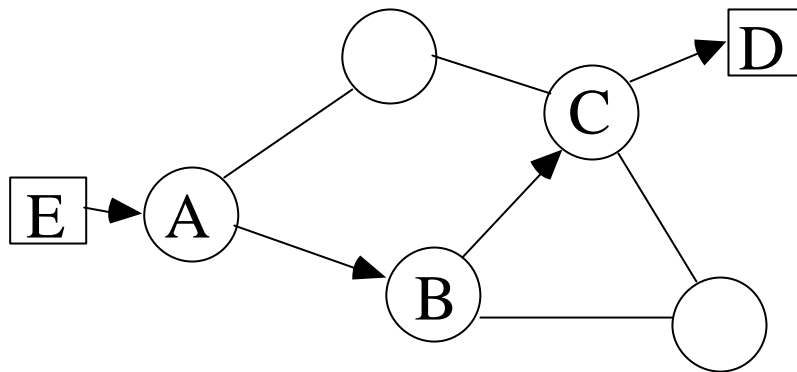
(exemple 12 bits).

. A l'ouverture on attribue un numéro libre.

. Chaque commutateur traversé peut modifier le numéro du circuit virtuel en lui attribuant un nom unique libre localement.

. Dans chaque commutateur pour assurer le routage il faut une table de correspondance entre les circuits entrants et sortants.

**Exemple de fonctionnement de l'adressage local des circuits virtuels *en mode unidirectionnel***



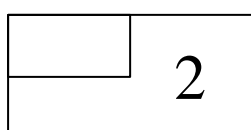
Entrants				Sortants			
E	2	B	1	A	1	C	3
Entrants				Sortants			

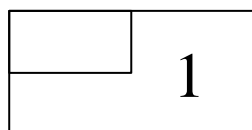
Entrants				Sortants			
Entrants				Sortants			

Table des circuits virtuels en A,B,C

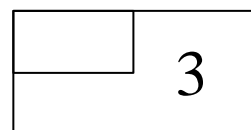
Évolution des entêtes de paquets (comportant la désignation du circuit virtuel)



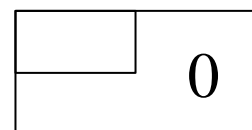
Paquets de E vers A



Paquets de A vers B



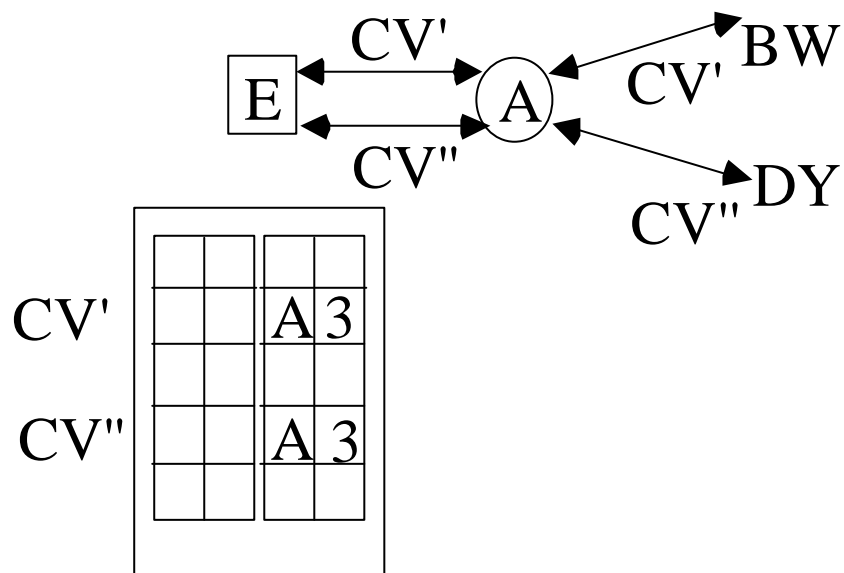
Paquets de B vers C



Paquets de C vers D

## Complément: utilisation en mode bidirectionnel des tables de circuits virtuels

- Les tables sont exploitées dans les deux sens (échanges en mode bidirectionnel).
- Deux sites A (un commutateur) et E (un hôte) échangent au même moment un appel.
- Par hasard E et A sélectionnent **le même** numéro libre (ici le 3):
  - . E a sélectionné 3 vers A pour un circuit virtuel en création CV' (sortant de E).
  - . A a sélectionné 3 vers E pour un circuit virtuel en création CV'' (entrant en E).



Pour CV' et CV'' quand un paquet arrive de A en E on ne sait pas à quel circuit il appartient.

## Solutions au problème d'adressage

### **Solution X25 : Notion de collision d'appel**

#### *Un conseil pour limiter le problème*

- E (l'hôte) doit émettre toujours un paquet d'appel sortant sur la voie logique de plus grand numéro libre.

- A (le commutateur de rattachement) doit émettre un paquet d'appel entrant sur le plus petit numéro de voie logique libre.

On peut quand même arriver à une collision d'appel.

#### *Une règle du protocole X25.*

En cas de collision d'appel (adresses identiques), le protocole doit détruire l'un des appels : l'appel entrant en E ( $\Rightarrow$  message d'erreur).

### **Solution Relais de trame**

Employer deux espaces de numérotation distincts pour les appels entrants et sortants.

## **2. LE PROBLÈME DE ROUTAGE**

## Position du problème

- A partir d'une adresse destinataire **choisir la meilleure direction de sortie.**

- Les noeuds mémorisent dans des tables de correspondance des informations du type:

destinataire final - prochain site à atteindre  
(contrôleur de voie physique à utiliser)

## Utilisation du Routage

Pour le mode circuit virtuel

-> A l'établissement du circuit virtuel.

Pour le mode datagramme

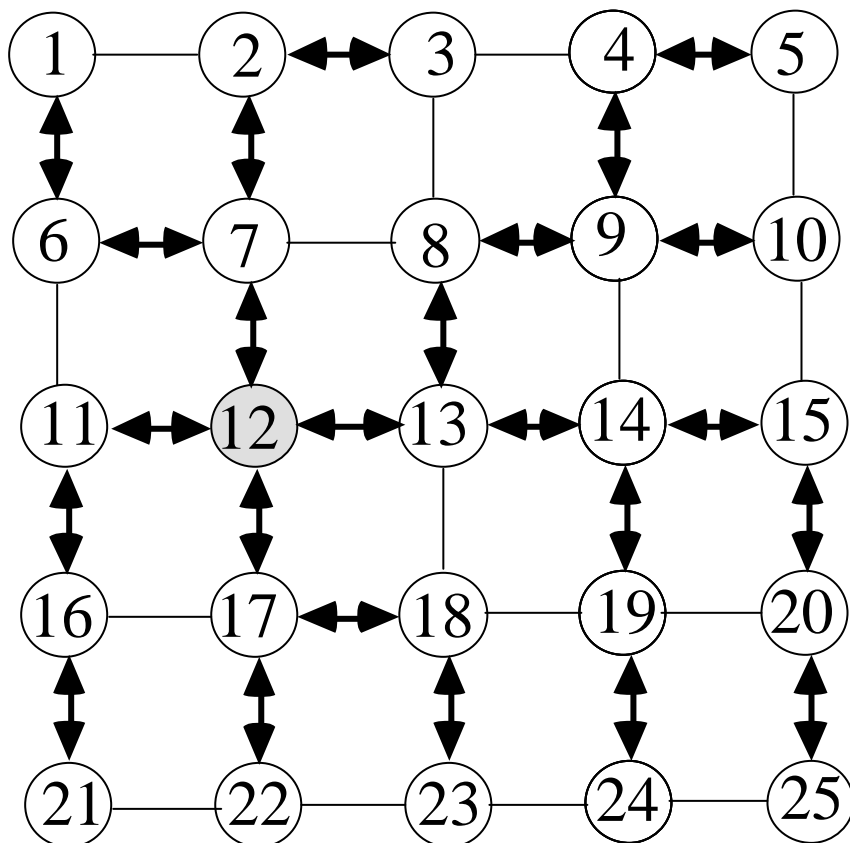
-> En permanence pour chaque paquet.



## Notion d'arbre couvrant (arbre de routage)

- Le **réseau de communication** est considéré comme un **graphe**.
- Chaque commutateur est considéré comme la **racine d'un arbre couvrant**.
- L'arbre définit les chemins pour atteindre tous les autres sites du réseau.

**Exemple d'un arbre couvrant**  
(pour le site 12 dans un réseau de type grille)



## **Arbre couvrant des plus courts chemins**

Construction d'un arbre couvrant qui optimise un critère de coût => **l'arbre couvrant des plus courts chemins**

### **Différents critères de coûts** (métriques, "metrics")

#### **- Longueur du chemin**

En nombre de commutateurs traversés.

#### **- Délai de traversée du chemin**

Délais d'attente dans les files des commutateurs, temps effectif de traversée.

#### **- Charge selon un chemin**

Mesure de la charge des commutateurs ou des voies empruntées.

#### **- Bande passante des voies empruntées**

9600b/s.

#### **- Fiabilité d'un chemin**

On envoie un message sur un chemin selon sa fiabilité.

#### **- Coût monétaire effectif**

Selon que l'on envoie un paquet sur une voie louée à coût marginal minime ou sur un réseau public "Transpac" avec tarification selon le volume => coût très différent.

# Algorithmes de routage

Le logiciel qui assure la gestion des tables de routage

## Notion de table de routage

- Pour tout site destinataire **déterminer dans une table le voisin immédiat** à qui transmettre un message (adjacent préféré).
- Éventuellement d'autres informations sont stockées
  - . **coût** du chemin,
  - . **délais** de garde sur ce chemin,
  - ....

**Exemple** : Table de routage du site 12 sur le réseau grille

Destinataire	1	2	3	4	5	6	...
Adjacent Préféré	7	7	7	13	13	7	...

### **Routages plats** ("flat routing")

Gestion d'un seul chemin par destinataire.

### **Routages multi-chemins** ("multi path")

Gestion de plusieurs chemins.

## Gestion des tables de routage

### *. Gestion statique (sans prise en compte de coûts)*

Définition statique de tables (compatibles).

### *. Gestion dynamique avec prise en compte de coûts stabilisés*

Principe d'optimalité (Bellman).

Si J est sur le chemin optimal de I à K alors le chemin de J à K est aussi optimal.

Justification de l'usage des tables.

### *. Gestion dynamique complète modification permanente des coûts*

- Possibilité de boucles pour certains messages.

Nécessaire convergence vers une solution stable si les coûts se stabilisent.

- Prise en compte des informations les plus récentes.

## Qualités d'un algorithme de routage

- **Correct** : Permet d'atteindre effectivement le destinataire.
- **Optimal**: Choix du meilleur chemin au sens du coût (Optimalité variant selon le critère de coût):  
Usager => Temps de réponse par message minimum.  
Réseau => Taux d'utilisation des voies, Taux d'utilisation des commutateurs.
- **Simple** : Comportement facile à décrire,  
Peu de types de messages échangés.
- **Efficace**: Ne consommant pas beaucoup de ressources (important si un hôte est également routeur)
- **Robuste** : Prévu pour tolérer les pannes du réseau
- **Adaptatif** : Évolutif en fonction de la charge (évite les phénomènes de congestion).
- **Stable** : Convergence rapide d'un routage dynamique vers de nouvelles tables lors de modifications Adaptativité réalisée "sans" oscillations.
- **Équitable** : Traite tous les sites de la même façon.

## Routage hiérarchique

### Pour un grand réseau

- . **Nécessité d'une très grande table**  
(pour déterminer l'adjacent préféré pour toutes les destinations).

- . **Temps de recherche important**  
(pour de nombreuses recherches par seconde)  
=> Solution: routage hiérarchisé à plusieurs niveaux.

### Exemple à deux niveaux

Division du réseau en régions ou domaines.

- . Deux types de routages pouvant relever de solutions différentes sont développées.

#### **Routage intra-domaine**

#### **Routage inter-domaine**

- . Chaque domaine réalise un routage interne intra domaine.

- . Pour chaque domaine un (ou plusieurs) commutateur sont spécialisés dans le trafic inter domaines (trafic externe).

# Classification des algorithmes de routage

## Adaptabilité en fonction de la charge (Critère le plus important)

- . Routage "non adaptatif",  
"prédéterminé", "statique"

Tables non modifiées automatiquement en fonction des conditions d'exploitation  
=> détermination manuelle.

- . Routage "adaptatif",  
"dynamique", "évolutif"

Tables modifiée périodiquement par programme en fonction de la charge et des pannes.

## Sûreté de fonctionnement

### **Routage "centralisé"**

( "dissymétrique" )

Un site spécialisé calcule les tables de routage => problèmes de panne et de surcharge du site de calcul.

### **Routage "décentralisé"**

( "réparti" , "symétrique" )

Toutes les stations calculent les informations de routage .

En fonction d'informations d'état locales ou globales

Échangées avec le reste du réseau (le voisinage immédiat ou plus)

=> Les solutions symétriques sont plus tolérantes aux pannes.

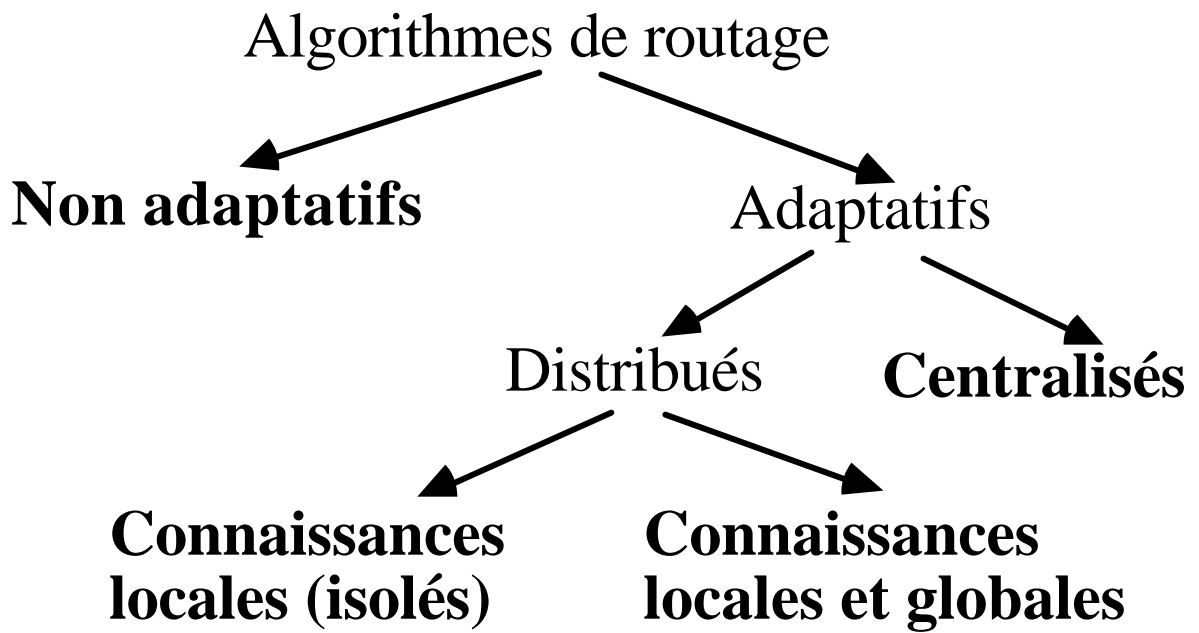
### **Routage hybride**

Pouvant utiliser à la fois un site central et un algorithme réparti

=> Selon les cas profiter des avantages d'un mode ou de l'autre.



# Catégories de routage étudiées



# Présentation détaillée de différents algorithmes de routage

## Routage par la source

("Paquets fléchés", "Source Routing")

- Chaque émetteur connaît son arbre de routage.
- Dans chaque paquet l'émetteur définit le chemin qu'il doit emprunter pour atteindre son destinataire.
- Les routeurs appliquent le chemin.

### Deux variantes

- . Définition **stricte** du chemin.
- . Définition d'un sous-ensemble de routeurs **qui doivent être traversés** (mais le routage peut en ajouter d'autres)  
Solution assez rare en usage unique  
(volume d'informations dans les messages).

Exemple: commutateurs de réseaux locaux IBM

## **Routage statique (routage fixe, non adaptatif)**

- Les règles de routage sont définies de manière **fixe, invariantes en fonction de la charge**.
- Elles sont implantées dans des **tables de routage stables modifiées rarement** (lors de l'insertion ou de la disparition de sites).
- Les tables sont établies en fonction de critères de topologie et de performance (évalués hors ligne) par le concepteur du réseau.

Volonté d'une optimisation globale et à long terme.

### **Deux approches possibles**

#### *- Routage simple*

=> On transmet une seule copie du message à un voisin.

#### *- Routage avec inondation*

=> On transmet plusieurs copies du message vers plusieurs adjacents.

## Routages statiques simples (à une seule copie)

### Table de routage à un seul choix ("routage plat")

Pour chaque destination on définit **un seul** adjacent préféré.

### Table de routage à plusieurs choix ("multi chemins")

Pour chaque destination on définit **plusieurs** adjacents possibles permettant d'atteindre le destinataire.

A la limite on peut définir tous les voisins comme possibles.

=> Permet de **tolérer** en opération **les pannes** des voies adjacentes sans devoir reconfigurer immédiatement la table.

## Solution "multi-chemins" Usage des différents choix prédéfinis

- Selon une **hiérarchie** pré définie.
  
  - En tirant **aléatoirement** l'une des voies possibles (Routage aléatoire "Random routing", "Stochastic routing").
  
  - En tirage **aléatoire** tenant compte des **capacités statiques** des voies adjacentes
- => Une voie possible deux fois plus rapide qu'une autre a deux fois plus de chances d'être empruntée.

## **Conclusion : routages statiques simples**

### **Avantages**

- Technique très simple à mettre en oeuvre

### **Inconvénients**

- Non adaptatif en cas de charge élevée.

### **Utilisation**

- Pour un réseau de petite taille de conception simple.
- Au démarrage d'une architecture de réseau le routage statique permet la définition d'un minimum de connaissances de routage avant mise en oeuvre d'une méthode avec apprentissage dynamique.

## **Routage statique par Inondation("Flooding")**

### **Principe de l'inondation générale ("Flooding")**

Pour tout paquet arrivé on transmet sur tous les adjacents d'un commutateur une copie.

Complément indispensable: une stratégie d'arrêt de l'inondation.

### **Principe de l'inondation sélective ("Selective Flooding")**

On transmet le paquet sur tous les adjacents d'un commutateur pris dans une liste.

- soit tirée aléatoirement dans l'ensemble des voisins.
- soit correspondant à des sites en direction du destinataire

Nécessité d'une stratégie d'arrêt de l'inondation (comme précédemment).

## Stratégie d'arrêt de l'inondation.

### **Solution 1 :** solution coûteuse

- On évite de renvoyer une copie du paquet au site dont on l'a reçu (puisque'il le connaît déjà).

- On note dans chaque message la liste des sites déjà visités pour ne pas leur renvoyer:

### **Solution 2 :** Solution coûteuse

- On note dans chaque copie le nombre de commutateurs traversés

La copie est détruite après traversée de  $n$  commutateurs (  $n >$  diamètre du réseau )

### **Solution 3 :** Solution efficace.

- Les paquets sont identifiés (estampille avec l'adresse émetteur et le numéro de séquence de l'émetteur):

La première copie reçue fait l'objet d'une inondation et les copies ultérieures sont détruites.



## Conclusion : routages avec inondation

### Avantages

- Technique **très simple** à mettre en oeuvre.
- **Très robuste**  
(détermine un chemin s'il existe)
- **Détermine le chemin le plus court**  
(devrait délivrer les paquets très vite sauf problèmes de surcharge).

### Inconvénients

- **Induit une charge élevée.**

### Utilisation de l'inondation

- Domaine des cas de pannes fréquentes ou importantes (militaire, reconstruction d'un réseau).
  - Référence de performance.
  - Protocoles à diffusion.

# Routage Adaptatif Centralisé

## Principe

- Existence d'un **site central** particulier (RCC "Routing Control Center")

=> Par exemple le site administrateur du réseau (chargé de la gestion des abonnés, de la facturation...).

-Émission par les commutateurs d'information de charge vers cet administrateur.

### . Périodiquement

### . Sur franchissement de seuils.

- Le central calcule les tables de routage (algorithme des plus courts chemins exécuté en mode centralisé) et les renvoie aux routeurs.

### **Variante: Serveurs de routes**

Plusieurs sites calculent des tables pour des routeurs qui ne font qu'appliquer les routes calculées.

# Conclusion routage adaptatif centralisé

## Avantages

- Simple à mettre en oeuvre.
- Solution optimale sur les valeurs utilisées
- Solution correcte sans boucles.
- Trafic stable d'informations échangées.
- Pas de calcul dans les commutateurs.

## Inconvénients

- Vulnérabilité de l'administrateur.
- Temps de calcul important.
  - => charge du central
  - => limitation de la fréquence de calcul
- Propagation des informations de charge
  - + Temps de calcul des tables
  - + Temps de retransmission des tables
  - => Usage de tables anciennes
- Trafic élevé au voisinage du central.

## Utilisation

- Très fréquente (réseaux commerciaux).

## **Routages Adaptatifs Distribués**

### **Routage local ou isolé**

#### **Routage de type pomme de terre chaude ("Hot potato routing")**

- Aucun site ne joue un rôle particulier dans l'algorithme (algorithme **adaptatif distribué**).
- Les commutateurs disposent uniquement d'informations **locales** (longueur de leurs files d'attente, taux d'occupation des voies).

#### **Pomme de terre chaude:**

##### **Version de base irréaliste**

- On transmet un paquet dans la direction qui est actuellement la moins chargée (par exemple la file d'attente la plus courte).

##### **Inconvénients** - Existence de boucles

- L'arrivée n'est pas garantie.

##### **Version réaliste**

- On transmet un paquet dans la direction la moins chargée dans un ensemble de choix possibles correspondant à la direction du destinataire.

# **Routage Adaptatif Distribué**

## **Apprentissage de connaissances globales**

### **Principe**

- Aucun commutateur ne joue un rôle particulier dans l'algorithme.
- Tous les commutateurs cherchent à disposer de connaissances globales sur le réseau pour effectuer un routage adaptatif optimal.

### **Exemple 1:**

#### **Apprentissage de coût des chemins**

#### **("Backward Learning")**

On apprend par des expériences successives des chemins possibles (et leurs coûts) pour atteindre un destinataire.

Exemple: commutateurs de réseaux locaux

- Utilisation d'une diffusion générale ou d'un routage aléatoire.
- On adopte le meilleur chemin.
- On recommence périodiquement: cas des modifications de topologie ou de charge.

**Exemple 2:**  
**Détermination de l'arbre couvrant des plus courts chemins par chaque noeud**  
**"Link-state", "Open Shortest Path First"**

## Principe

- Chaque routeur détermine son arbre des plus courts chemins.
- Pour cela il doit connaître l'état des liaisons de tous les autres.
- Il applique un algorithme de plus courts chemins centralisé.  
=> Solution plutôt coûteuse : applicable intra-domaine.

## Fonctionnement par information mutuelle

- Chaque routeur envoie à tous les autres l'état de ses liaisons ("link state") lorsqu'il le juge nécessaire
  - Chaque routeur dispose d'un état global de la topologie et des coûts sur son domaine.
  - Il calcule son arbre de routage.
- Exemple: routage OSPF Internet

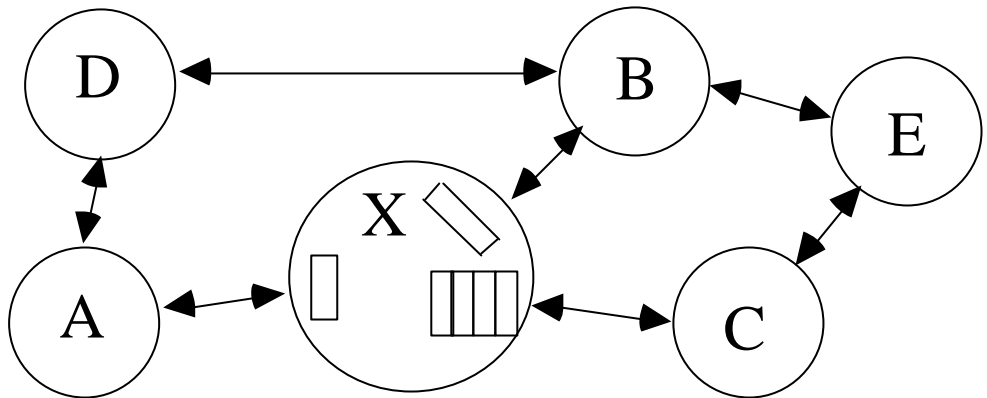
### **Exemple3: Détermination des tables de routages par échange de vecteurs de distances avec les voisins ("Distance vector", RIP)**

Autres appellations : "Algorithme de Mac Quillan" ou "Bellman Ford Fulkerson réparti"

#### **Principe**

- Chaque routeur gère deux tables:
  - . une table des coûts de transit vers ses voisins immédiats déduites d'infos locales.
  - . une table de routage avec coûts vers tous les autres sites du réseau obtenue par échange avec les voisins immédiats.
- Chaque routeur émet périodiquement cette table de routage vers ses voisins et reçoit de ses voisins leurs tables de routage.
- Il recalcule sa propre table en additionnant:
  - . le coût pour atteindre un voisin immédiat
  - . le coût de ce voisin à tous les autres sitesIl obtient le coût pour atteindre tous les sites en passant par ce voisin.
- L'adjacent préféré est le voisin de coût minimum pour atteindre un destinataire.

# Exemple de fonctionnement



## Avant optimisation: état du site X

Coût Local	A	B	C	Coût Distant	D	E
X	26	20	18	A	64	130
				B	56	115
				C	45	210

Pour aller de X à D ou de X à E  
 Par A  $26 + 64 = 90$   $26 + 130 = 156$   
 Par B  $20 + 56 = 76$   $20 + 115 = 135$   
 Par C  $18 + 45 = 63$   $18 + 210 = 228$

## Après optimisation: table de routage X

X	A	B	C	D	E
Adjacent	A	B	C	C	B
Coût	26	20	18	63	135



## Analyse du routage par vecteur de distance

### Avantages

- Algorithme complètement réparti
- Simple à développer.

### Inconvénients

## Problème de la fréquence d'échange des tables

. Pour limiter le trafic et les traitements  
=> **Peu d'échanges.**

. Pour que les modifications soient prises en compte vite  
=> **Échanges fréquents.**

Les bonnes nouvelles se propagent vite:

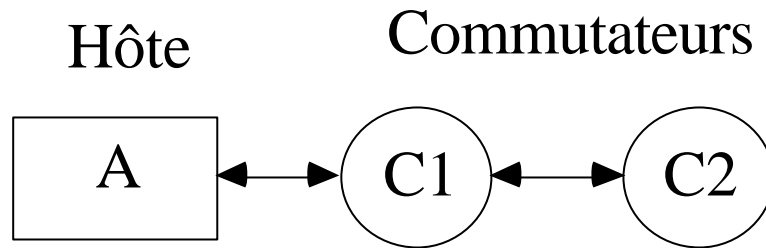
**Le débit s'améliore.**

Les mauvaises nouvelles circulent lentement.

**Apprentissage** d'un événement dans tout le réseau => D étapes d'échanges de tables ou D est le diamètre du réseau.

Exemples de fréquence: 0,640 s , 30 s.

## Problème du comptage vers l'infini



- Métrique utilisée: le **nombre** de sauts:  
 $\text{coût}(A-C1) = 1$  ,  $\text{coût}(A-C2) = 2$
- **A-C1** tombe en **panne** :  $\text{coût}(A-C1) = 8$
- Au calcul suivant **C2 indique à C1** qu'il a un chemin vers A **de longueur 2** par C1.
- C1 peut considérer (à tort) qu'il a **retrouvé un chemin vers A** de longueur 3 par C2.  
**=> Les paquets bouclent.**
- A chaque calcul les coûts augmentent de 1.  
Effet dit de **comptage vers l'infini**.

# Solutions au problème de comptage vers l'infini

## Solution de base

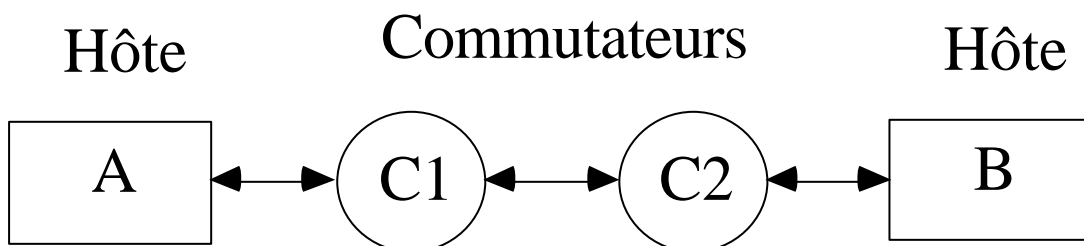
. Introduire une borne dans le coût des chemins.

. Si un chemin a un coût plus élevé que la borne on considère qu'il est coupé (coût infini).

## Autre solution

- **Limiter le volume des tables échangées** en ne renvoyant pas une information dans la direction d'où elle vient ("split horizon")

=> Échange partiel de tables



- C1 indique à C2 qu'il a une route vers A.
- C2 ne retransmet pas cette route à A (puisque'elle vient de A).
- On évite ainsi le comptage vers l'infini.

## **Conclusion: protocoles à vecteurs de distance**

### **Très nombreuses utilisations**

- Internet: Protocoles RIP, EGP  
"Routing Information Protocol".
- XNS "Xerox Network System"  
Protocole GWINFO.
- Apple talk RTMP  
"Routing Table Maintenance Protocol"
- Novell, 3COM, Ungermann-Bass, Banyan.

Dans Internet l'utilisation diminue depuis 1990 au profit de OSPF.

# Routage Adaptatif Hybride (Centralisé et distribué) Algorithme Delta de Rudin

## *Niveau global*

Un site spécialisé (administrateur)

- **Reçoit** des informations de charge,
- **Calcule** pour tous les routeurs et toutes les destinations plusieurs chemins possibles de coûts croissants à partir de l'optimum.
- **Retransmet** les choix possibles et leur coût.

## *Niveau local*

- Un commutateur a une valeur de seuil  $\delta$ .
- Compare les coûts à  $\delta$  près.

Si la différence de coût est inférieure à  $\delta$  deux routes sont considérées comme équivalentes.

=> le commutateur peut envoyer un message sur un adjacent ou un autre.

Le commutateur choisit entre des routes équivalentes selon des critères de charge locale (coût pour atteindre l'adjacent).

$\delta = 0$  l'algorithme est centralisé

$\delta = \text{infini}$  l'algorithme est réparti (local)

## Conclusion Routage

### Problème important de l'algorithmique répartie

- Très nombreuses solutions proposées.
- Algorithmes généraux de construction d'arbres couvrants ou d'arbres couvrants des plus courts chemins dans un réseau.
- Solutions particulières pour des types de réseaux particuliers:
  - . routage sur des réseaux à topologie régulière (grille hypercube)
  - . routage pour des réseaux téléphoniques.

### **3. LE PROBLÈME DE CONTRÔLE DE CONGESTION**

## Introduction: la congestion

### Situation de surcharge:

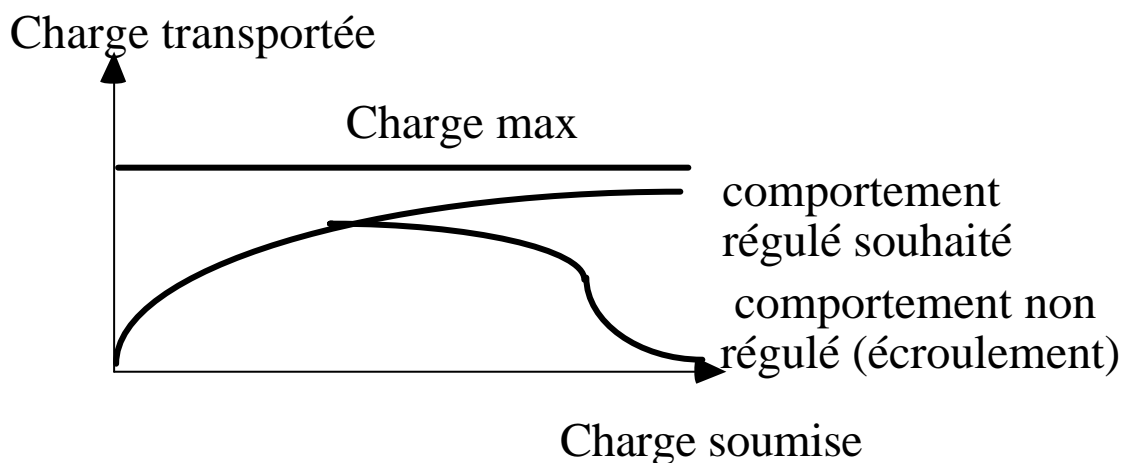
Trop de paquets dans le réseau (ou dans un région du réseau).

### Situation de congestion:

.**Augmentation des délais** de transmission  
=> non conformité à la définition de service

### .Écroulement

Le réseau ne transporte plus le trafic usager



### Pourquoi

- **Destruction abusive de paquets** pour désengorger les noeuds (il faut retransmettre).
- **Protocole de traitement de surcharge** (qui implique de plus en plus d'échanges).



## **Traitement préventif par limitation du trafic d'entrée**

### **Limitation du trafic de chaque station**

- . Selon un contrat défini au préalable entre le client et le prestataire
- . Pour éviter une surcharge non contrôlée.
- . Également pour découpler le trafic soumis par chaque usager du débit de sa voie physique de rattachement.

### **Fonctionnement**

- L'usager selon la qualité de service souscrite doit respecter des règles de débit soumis.
- La définition doit être suffisamment fine pour éviter que l'usager ne soit pas trop contraint => Nombreux paramètres possibles
  - . Nombre moyen de paquets par unité de temps
  - . Amplitude des rafales autorisées...
- Le prestataire doit doter chaque accès d'un contrôleur de débit d'entrée ("leaky bucket").

## **Conclusion: Limitation du trafic d'entrée**

### **Avantages**

On évite des comportements inattendus de certains usagers.

### **Inconvénients**

Si la limitation du volume global entrant est trop importante:

=> sous-réservation des ressources

Si la limitation est trop légère.

=> la saturation et la congestion peuvent apparaître.

### **Utilisation**

- Relais de trames ("Frame Relay")
- ATM

### **Conclusion**

- Une technique introduite pour limiter les difficultés.

- Ne constitue pas une solution qui empêche l'installation de la congestion.

## **Traitement préventif de la congestion** **Pré allocation des ressources**

### **Cas des circuits virtuels**

- En fonction du contrat de débit négocié.
- Au moment du passage du paquet d'appel
- Réserve dans tous les commutateurs traversés des ressources pour l'acheminement correct du trafic (tampons, bande passante).
- Si la réservation est impossible le circuit n'est pas accepté.
- Si la réservation a pu avoir lieu, le trafic doit pouvoir toujours s'écouler.

### **Avantage**

- Si la gestion est rigoureuse la congestion est impossible.

### **Inconvénient**

- Les usagers n'utilisent pas les ressources qu'ils réservent => le fournisseur du service est tenté de faire de la sur-réservation pour optimiser les ressources dont il dispose.

**Implique l'acceptation d'un certain niveau de congestion**

# **Traitement préventif de la congestion**

## **Limitation globale du trafic soumis**

### **Contrôle Isarithmique**

## **Principe**

- Chercher à éviter l'engorgement en gardant constant le nombre de paquets en circulation dans un réseau.

=> Il existerait un trafic supportable globalement par le réseau.

- Pour cela faire circuler des droits d'accès: la possession d'un droit autorise un site à faire transiter un paquet.

## **Inconvénients**

- Possibilité d'engorgements locaux (sauf à introduire dans le réseau un nombre de droits dérisoire) => la congestion partielle subsiste.

- Possibilité de famines locales (manque de droits pour émettre dans une région).

- Surcharge due à la circulation des droits

- Perte progressive des droits par perte de messages ou par panne des commutateurs.

## **Traitement curatif de la congestion**

### **Destruction de paquets**

#### **Engorgement d'un commutateur**

Le site ne traite plus les paquets entrants.

- Insuffisance de **mémoire** pour stocker.
- Insuffisance de **puissance de calcul** et de **débit** des voies pour vider la mémoire.

#### **Destruction de paquets**

- Incontrôlée dans les tampons d'entrée.
  - Selon une politique contrôlée
    - . Après avoir exploité les informations de libération de ressources (acquittements).
    - . Pour maintenir un trafic "plus prioritaire au détriment d'un autre (gestion séparée des files d'attente, des circuits virtuels).
- Notion de bit de priorité à la destruction.

#### **Conclusion**

- La solution la moins bonne pour des données informatiques (il faut retransmettre ensuite).
- Si la destruction est acceptée ou inévitable
  - Acceptation du taux d'erreur introduit.
  - . comparable aux erreurs de transmission

# Traitement curatif de la congestion

## Paquets de surcharge ("Choke Packets")

### Principe

- Sur franchissement de **seuil** d'alerte demande aux voisins de limiter ou de suspendre les communications:

- . **par des paquets spécifiques de surcharge** (trafic en plus)

- . **par des bits de surcharge** insérés dans les paquets courants ("piggybacking").

Exemple: ATM bit EFCN

"Explicit Forward Congestion Notification"

- L'indication de surcharge

- . s'applique à un **CV** ou une **voie** physique

- . peut **s'arrêter aux voisins** du site surchargé ou être **propagée jusqu'aux hôtes générant le trafic** d'entrée.

**Avantages** Peut réellement traiter une situation de congestion.

**Inconvénients** En surcharge lenteur de circulation des informations de surcharge.

## Le contrôle de congestion basé crédit et le contrôle basé débit

Sur quel critère traiter la congestion: espace mémoire occupée ou débit des communications

**Crédits** : Une information qui concerne l'espace mémoire disponible (le nombre de tampons libres).

**Algorithmes basés crédits:**  
=> faire circuler des crédits

**Débits** : Une information qui concerne le nombre de paquets ayant circulé par unité de temps dans un passé récent.

**Algorithmes basés débits:**  
=> faire circuler des paquets de surcharge lorsqu'un indicateur de débit est positionné.  
Avantage des solutions basées débit sur les solutions basées crédit dans le cas des réseaux à haut débit: l'espace mémoire (très important) reste inoccupé trop longtemps).

## Contrôle de flux

- Ne pas confondre les notions voisines de
  - . **contrôle de flux**
  - . **contrôle de congestion**
  - . **limitation du débit d'entrée**
- En anglais "Flow Control" utilisé dans les trois sens.

## Contrôle de flux

Rétroaction d'un récepteur sur un émetteur pour qu'il puisse traiter les messages reçus.

Au niveau réseau : contrôle de flux d'hôte à hôte par des mécanismes identiques à ceux du niveau liaison

- **Utilisation des mécanismes de fenêtres** glissantes (d'amplitude maximale fixe ou de crédits variables).
- **Utilisation des demandes de suspension temporaire** de transmission.



## **Conclusion**

### **Problèmes Généraux du niveau Réseau**

Un domaine central des architectures de réseau.

En renouvellement important avec l'arrivée des réseaux haut débit sur fibre optique (gigabit).

- pour l'interconnexion des réseaux locaux
- pour le développement du RNIS large bande.

# 4

## **LE ROUTAGE INTERNET**

## Introduction

**Objectif** : Atteindre un destinataire dans un très **grand réseau** en masquant la traversée d'une série de **réseaux intermédiaires**.

Deux types de systèmes

### Hôtes "Hosts"

Un hôte dispose d'une table de routage simplifiée.

Il ne relaie pas de messages.

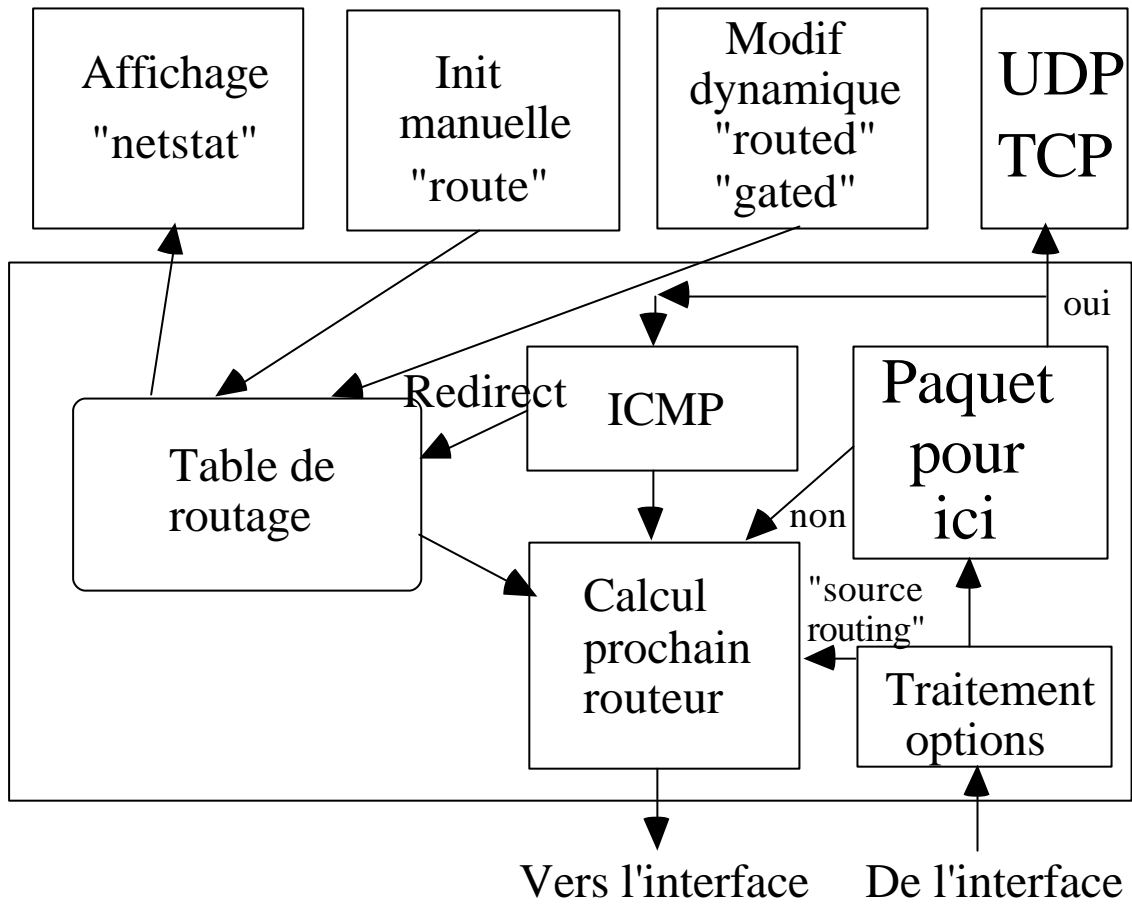
### Routeurs, Commutateurs "Routers" "Gateways"

. Un routeur peut-être n'importe quel ordinateur avec IP à condition qu'il soit configuré en routeur

=> souvent un matériel spécialisé

. Un routeur retransmet un message d'une interface entrante à une interface sortante s'il dispose des informations suffisantes sinon il note le message 'non délivrable'.

# Organisation générale du routage IP



## Structure de la table de routage

Exemple de la structure sous UNIX

### **. Adresse IP destination**

Ce peut-être une adresse d'hôte ou une adresse de réseau (la zone Host-id est à zéro).

### **. Adresse du prochain routeur**

A emprunter pour atteindre la destination.

### **. Indicateurs ("flags")**

U chemin opérationnel, G chemin vers un routeur, H chemin vers un hôte, D chemin créé par une redirection, M chemin modifié par une redirection.

### **. Nombre de références**

Nombre de connexion utilisant le chemin (exemple une connexion TCP, un telnet).

### **. Nombre de paquets envoyés**

### **. Interface**

Sur laquelle envoyer le paquet ("device").

## Exemple de la commande UNIX "netstat"

```
% netstat -r
```

```
Routing tables
```

Destination	Gateway	Flags	Refcnt	Use	Interface
163.173.128.0	asimov	UH	0	0	xna4
localhost	localhost	UH	34	2936	lo0
default	internet-gw	UG	51	424780	xna4
163.173.128	asimov	U	80	619003	xna4

Comporte en général 3 entrées pour un hôte:

- La boucle locale

(pour les messages qui ne sortent pas du site)

- L'accès au réseau ethernet de l'hôte.

- L'accès à un routeur par défaut qui permet de passer sur l'internet

```
pouchan% netstat -r
```

```
Routing tables
```

Destination	Gateway	Flags	Refcnt	Use	Interface
localhost	localhost	U	16	1516	lo0
163.173.136.0	pouchan	U	5	21407	le0
default	cisco-turbigo	UG	5	6153	le0

## **Fonctionnement de base du routage**

### **Etapes d'une opération de routage d'un hôte**

- Si le site à atteindre est connecté directement au site courant (par une liaison point à point ou en réseau local)  
=> le message est envoyé directement.
- Sinon l'hôte dispose d'un routeur par défaut à qui il envoie tous les datagrammes qu'il ne peut acheminer.

### **Etapes d'une opération de routage d'un routeur**

- Recherche d'une destination correspondant à celle visée.
- Recherche d'une entrée réseau où se trouverait le site visé.
- Recherche d'une entrée de type défaut.

## Administration de la table de routage

De très nombreuses possibilités de routage statique ou dynamique sont disponibles.

### Initialisation Statique

- **Commande ifconfig (gestion des pilotes de coupleurs réseaux)**

**"ifconfig" configure les paramètres du pilote**

. A chaque définition d'une interface la table de routage est initialisée automatiquement en conséquence.

**Exemple :** /etc/ifconfig ie0 iris up

- déclare un coupleur ethernet ie0 (ie pour Intel)
- sur l'hôte "iris" (déclaré dans /etc/hosts)
- contrôleur actif (up)

### Autres options

"netmask" : définition du masque de sous-réseau.

"interface" Liste des paramètres de l'interface

"interface" **down** Arrêt d'un interface



## - Commande route

. Si l'on veut déclarer explicitement un chemin vers un site distant.

**route** ajout destination routeur metrique

Exemple:

```
route add -net 192.56.76.0 netmask  
                255.255.255.0 dev eth0
```

. Les chemins initialisés statiquement sont contenus dans un fichier de configuration :  
BSD /etc/netstart , SVR4 /etc/inet/rc.inet , AIX  
/etc/rc.net, SUNOS /etc/rc.local, SOLARIS 2  
/etc/rc2.d/S69inet

# Modification des tables par découverte de chemins

## Redirection par ICMP

- ICMP se charge d'acheminer les messages d'erreur pour IP (en particulier le fait qu'un destinataire soit inaccessible).

- Utilisation des diagnostics ICMP pour améliorer le routage

Ex: . Un routeur A envoie un message à un routeur B.

. B l'envoie à C mais s'aperçoit qu'il utilise pour cela la liaison vers A.

. B indique par un message ICMP à A cette anomalie: envoyer directement à C.

## Messages ICMP de maintenance des tables

- ICMP peut diffuser des demandes de routes (en diffusion totale ou mieux en diffusion sur groupes).

"router solicitation message"

- Les routeurs à l'écoute répondent.

"router advertisement message"

## **Routage dynamique**

Le **routage dynamique** ne change rien à la façon dont les tables de routages sont utilisées dans la couche IP.

Les informations qui sont contenues dans les tables sont modifiées au fur et à mesure par des processus (démons).

### **Routage en deux classes**

**Domaine** (AS "Autonomous Systems")

**Inter domaine**

Un domaine correspond à un ensemble de sites, administrés par une seule et même entité (grande entreprise, campus).

### **Protocoles de routage intra-domaine**

(IGP "Interior Gateway Protocol")

- . RIP " Routing Information Protocol"
- . OSPF " Open Shortest path First"

### **Protocoles de routage inter-domaine**

(IRP "Interdomain Routing protocol")

- .EGP "Exterior Gateway Protocol"
- .BGP "Border Gateway Protocol"

# RIP Routing Information protocol (RFC 1028)

- **Solution par échange périodique de tables de routages ("Distance vector")**

- **Initialisation**

.Emission d'une requête de demande de table sur toutes les interfaces avec une liste de destination (code commande 1).

. Préparation de la réponse

S'il y a une route pour une destination

=> retour de la métrique

Sinon => métrique = 30 (valeur infinie)

. Utilisation de la réponse (commande 2)

- **En mode établi (message request)**

. Mise à jour périodique : émission vers tous les voisins systématiquement toutes les 30 secondes. Si une route n'a pas été rafraîchie pendant 3 minutes elle est portée à 30 (infini pour destruction)

. Mise à jour sur événement (changement de métrique).

## Informations complémentaires RIP

. Métrique: le nombre de noeuds traversés ("hops").

. Protocole utilisé: UDP pour échanger ses informations avec les autres noeuds de routage.

. Nom du démon: "Routed"

.RIP V2 RFC 1388: extensions dans des champs inutilisés par RIP (diffusion, identification de domaine, indicateurs EGP)

**Sous UNIX commande pour obtenir les informations de routage** d'un routeur distant (code commande poll 5):

```
ripquery -n "nom de routeur"
```

# OSPF Open Shortest Path First (RFC 1247)

Remplacant progressivement RIP après 1990  
insuffisant pour les grands réseaux.

## Principes généraux

Collecte par chaque routeur de l'état des  
liaisons adjacentes (solution "link state").

Diffusion en inondation de ces états.

Calcul par tous les routeurs des tables de  
routage optimales par l'algorithme de Dijkstra  
("Shortest Path First").

<b>Type de message</b>	<b>Description</b>
Hello	Découverte des voisins immédiats
Link state update	Diffusion aux voisins
Link state ack	Acquittement réception
Database description	Annonce des états dont dispose un routeur
Link state request	Demande informations à un routeur

## Compléments OSPF

- Pour maîtriser la complexité des grands réseaux gestion par OSPF à l'intérieur d'un domaine ("autonomous system")
  - de régions ("areas")
  - dont une plus particulière permet de connecter les autres ("backbone" "area 0").
- On calcule des routes intra-régions, inter régions et inter domaine
- Usage de métriques pouvant être très variées
  - . Débit, délai aller-retour, ...
  - . Une métrique peut-être attribuée par type de service
- Protocole IP pour l'échange des informations
- Calcul de plusieurs routes possibles par type de service.
  - => Si deux routes sont de coût équivalent: distribution de charge.
- Nom du démon UNIX: "Gated"

# **BGP Border Gateway Protocol (RFC 1267, 1268)**

## **Principes**

Contrôler l'acheminement des paquets dans l'internet (interdire du trafic en transit , orienter le trafic, ...).

=> Politique spécifique aux grandes organisations.

Solution par échange de tables de routages comme RIP ("distance vector", Mac Quillan)

On gère pour chaque destination le chemin

- Trois types de domaines (identifiés par des entiers 16 bits):

- . domaine souche ("stub"):

  - un seul lien avec l'exterieur

- . domaine de transit ("transit AS") :

  - plusieurs liens avec l'extérieur, et autorise le passage d'informations extérieures à son trafic local (transit)

- . domaine multi-liens "multi homed AS":

  - plusieurs liens avec l'extérieur, mais n'autorise pas le passage.

- Utilise TCP pour échanger les informations.



## Conclusion Routage

### **Problème principal : L'agrégation de routes, la hiérarchisation**

- Le routage IPv4 est devenu suffisamment hiérarchique avec les notions de réseaux, sous-réseaux, les AS etc mais le réseau Internet est très grand et en développement rapide.

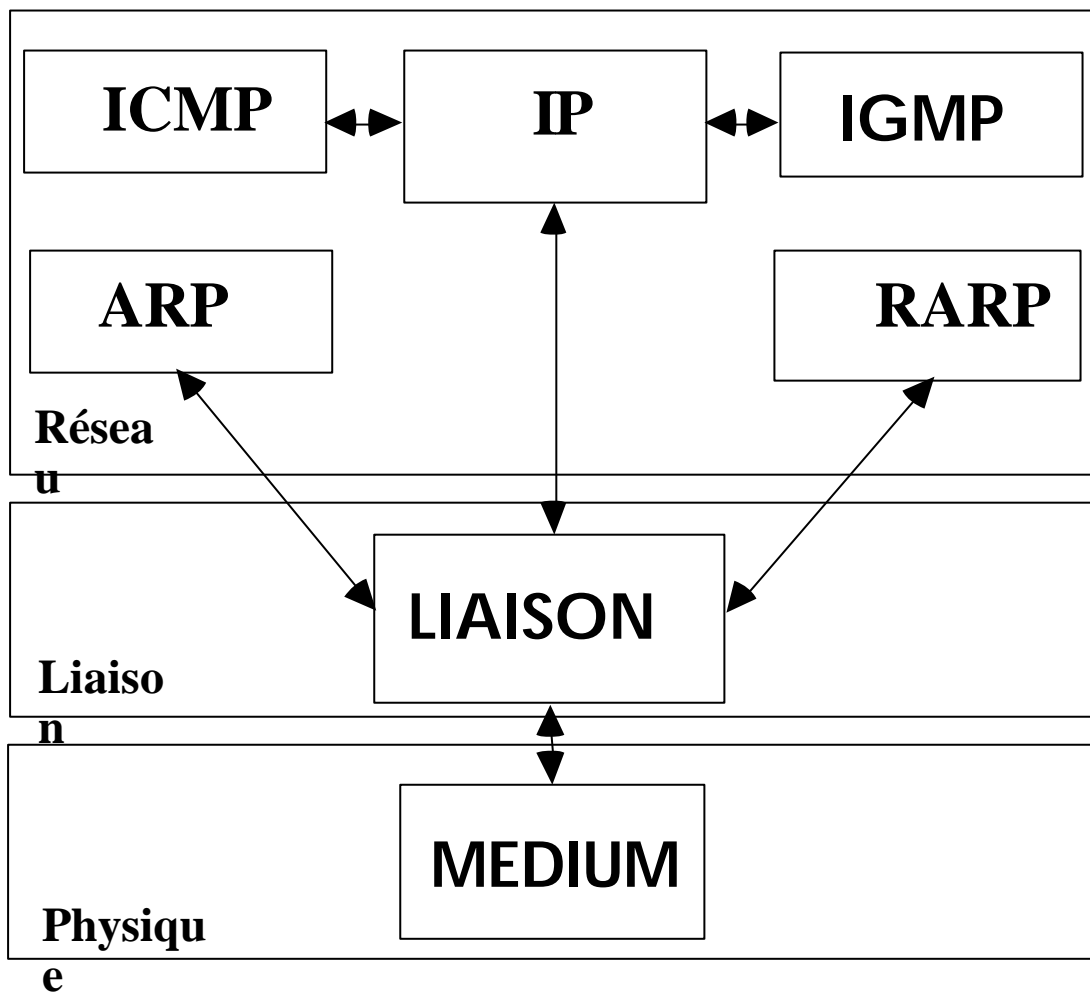
#### Introduction de IP V6:

Les algorithmes de routages sont basés exactement sur les mêmes principes (OSPF, BGP) mais le problème des adresses est résolu.

=> Rôle de plus en plus important des opérateurs, des grandes organisations et des fournisseurs de service qui administrent le routage pour des domaines importants.

**LES PROTOCOLES  
COMPLÉMENTAIRES  
DE IP**

# Relations entre les différents protocoles en version 4



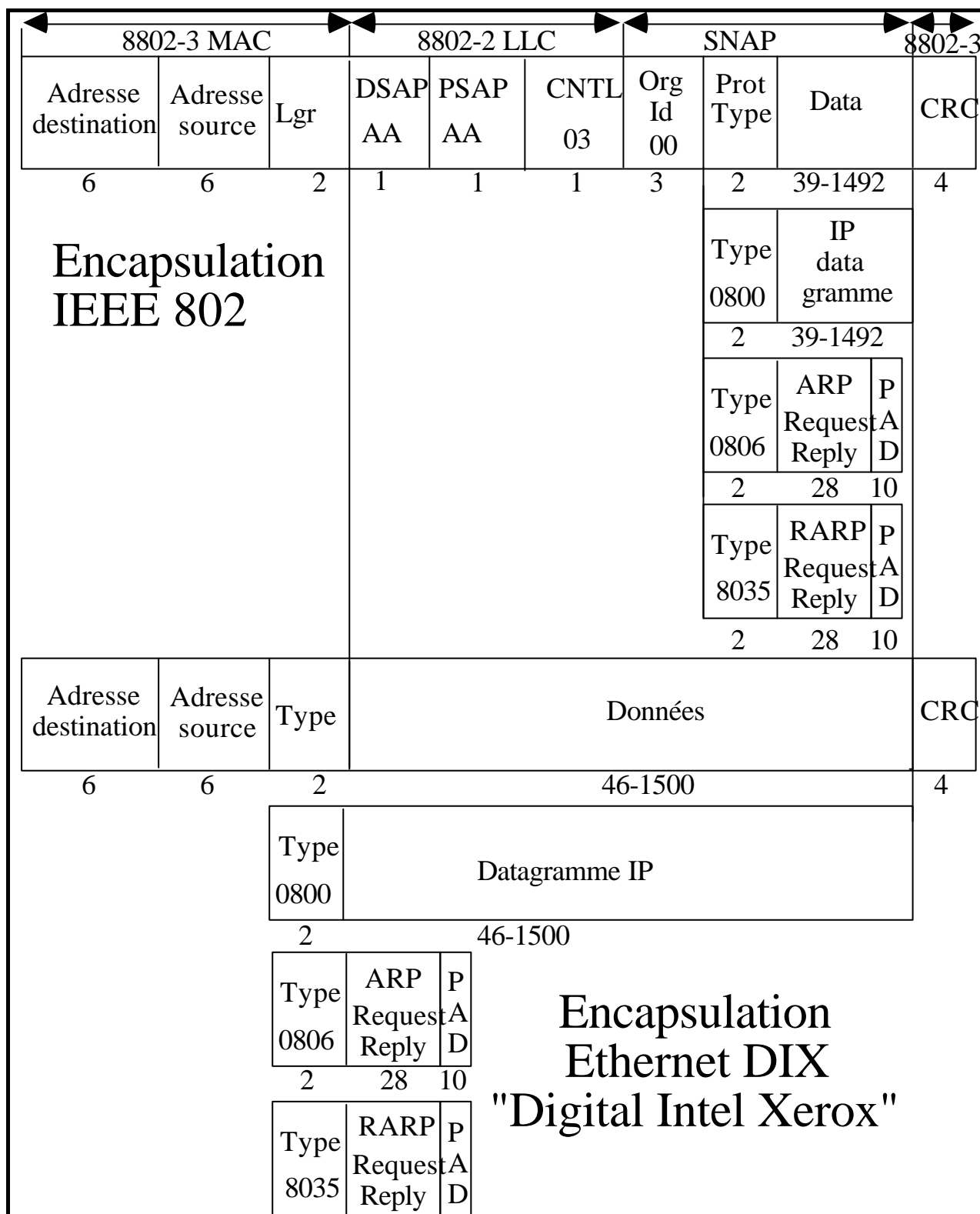
## **Liaison**

### **Réseaux locaux Protocoles LLC/SNAP**

#### **Principales fonctions**

- Encapsulation de paquets IP dans des trames format ethernet DIX ou IEEE 8802.
- Supporter conjointement sur le même réseau local différents protocoles de niveau 3 (IP, IPX, DECNET, CLNP, AppleTalk, ...)
- Niveaux Liaison IEEE 8802-2 **LLC**  
**"Logical Link Control"**  
3 octets adresse source, adresse destination, zone contrôle => par convention AA AA 03
- Encapsulation multi protocole **SNAP**  
**"Sub-Network Access Protocol"**  
5 octets dont
  - 3 octets "Organisation Ident" 000000
  - 2 octets " Protocol Type" Ex IP 0x0800

# Schéma des trames



# Protocoles de resolution d'adresse ARP/RARP

## Objectif

Correspondance biunivoque entre adresses liaison (adresses MAC) et adresses réseau (adresses IP):

**Adresses internet**  $\Leftrightarrow$  **Adresses ethernet**

@I<sub>A</sub>

P<sub>A</sub>

Etablir la liaison entre les deux types d'adresses.

## Solutions statiques

- Gérer dans chaque site une table statique  
=> modifier la table à chaque altération.
- Essayer de fabriquer des adresses IP et des adresses internet qui se déduisent l'un de l'autre par un calcul direct  
Perte de capacités d'adressage en V4.  
Possible en V6.

## Solution dynamique ARP/RARP

- Existence de sites serveur qui connaissent la relation d'adressage.
  
- Utiliser un protocole client-serveur pour déterminer dynamiquement les relations entre adresses (réseau souvent reconfiguré arrêt installation/retrait de stations de travail)
  
- Pas de connaissance des adresses de serveur: utilisation du mode diffusion des réseaux locaux.

**Problème 1** : Connaître une adresse ethernet connaissant l'adresse IP

=> **ARP**

**Problème 2** : Connaître une adresse IP connaissant l'adresse ethernet

=> **RARP**

## Fonctionnement ARP

A ( $@I_A, P_A$ ) doit dialoguer avec B ( $@I_B, P_B$ )  
Mais A ne connaît que l'adresse Internet  $@ I_B$   
 $\Rightarrow$  A veut connaître l'adresse physique  $P_B$

### Mécanisme de Cache d'adresse

Il existe une table qui conserve les résolutions d'adresses effectuées.

### Fonctionnement de la recherche

- (1) **A cherche dans son cache local** l'adresse ethernet: si succès fin du protocole et communication avec B.
- (2) **Si échec** diffusion d'un paquet spécial contenant  $@ I_B$
- (3) **Tous les sites du réseau reçoivent le paquet ARP** et comparent l'adresse internet proposée avec leur propre adresse  
 $\Rightarrow$  B répond seul en envoyant  $P_B$ .
- (4) **A entre l'adresse B dans sa table** et dialogue avec B.



## Compléments ARP

- B mémorise l'adresse physique PA de A.
- Tous les hôtes mémorisent l'adresse PA de A (lorsqu'elle circule en diffusion).
- A la mise en marche une station diffuse son adresse IP.
- Les entrées de la table sont détruites après une période dépendante de l'implantation (gestion dynamique de cache) pour tenir compte des modifications de l'architecture.

## Exemple de fonctionnement ARP UNIX

```
tulipe::/users/ensinf/gerard _23 /usr/sbin/arp -a
```

```
Net to Media Table
```

Device	IP Address	Mask	Flags	Phys Addr
le0	prithivi	255.255.255.255		08:00:20:04:3a:dd
le0	mac-florin.cnam.fr	255.255.255.255		00:00:94:22:ac:74
le0	cisco-for-turbigo	255.255.255.255		aa:00:04:00:1f:c8
le0	kendatt	255.255.255.255		08:00:20:7d:72:08
le0	savitri.cnam.fr	255.255.255.255		08:00:20:04:3a:04
le0	pouchan.cnam.fr	255.255.255.255		08:00:20:7d:ef:4d

```
< Longue liste de toute la table >
```

```
tulipe::/users/ensinf/gerard _25 /usr/sbin/ping rita
```

```
rita.cnam.fr is alive
```

```
tulipe::/users/ensinf/gerard _26 /usr/sbin/arp -a
```

```
Net to Media Table
```

Device	IP Address	Mask	Flags	Phys Addr
le0	prithivi	255.255.255.255		08:00:20:04:3a:dd
le0	mac-florin.cnam.fr	255.255.255.255		00:00:94:22:ac:74
le0	cisco-for-turbigo	255.255.255.255		aa:00:04:00:1f:c8
le0	kendatt	255.255.255.255		08:00:20:7d:72:08
<b>le0</b>	<b>rita.cnam.fr</b>	<b>255.255.255.255</b>		<b>08:00:20:12:bd:d5</b>
le0	savitri.cnam.fr	255.255.255.255		08:00:20:04:3a:04
le0	pouchan.cnam.fr	255.255.255.255		08:00:20:7d:ef:4d

```
< Longue liste de toute la table >
```

## **RARP**

### **"Reverse Address Resolution Protocol"**

#### **Déterminer une adresse IP avec une adresse Ethernet**

Exemple: machine qui boot.

En ROM adresse de coupleur de boot

=> Cas normal: boot à partir de disque.

=> Cas sans disque: boot à partir du réseau (coupleur boot = coupleur ethernet).

Nécessité pour utiliser un transfert de fichier simple (TFTP) de connaître l'adresse IP locale ou envoyer le binaire.

#### **Fonctionnement RARP**

=> La station qui amorce diffuse son adresse Ethernet sur le réseau local.

=> Un serveur RARP (nécessaire sur chaque réseau) lui renvoie son adresse IP.

Retransmissions (nombre limité).

Plusieurs serveurs RARP.

**Z En IPv6 ARP et RARP sont intégrés à ICMP.**

## **ICMP**

### **"Internet Control Message Protocol"**

- **ICMP réalise différents échanges** de messages entre stations pour véhiculer ou acquérir des informations.

Environ 15 types de messages échangés.

#### **Requête-Réponse d'obtention de masque (d'un sous-réseau)**

Pour une station sans disque qui souhaite le connaître lors de son initialisation.

#### **Requête-Réponse d'obtention de l'heure d'un site distant**

Pour développer une synchronisation d'horloge?

#### **Acheminement de messages d'erreurs**

Durée de vie dépassée => paquet détruit  
Site inaccessible ("port unreachable error") 16 diagnostics particuliers d'échec.

## Requête-Réponse permettant de tester si une station fonctionne correctement

**Paquet ECHO\_REQUEST**  
**Réponse ECHO\_REPLY**

Commande Unix:

**ping** “adresse IP” ou “nom de station”

Réponse : **"nom de station" is alive**

Très utile pour tester si la couche IP d'une station est active

**=> Machine opérationnelle ou non.**

**=> Détection d'une machine distante surchargée.**

Autre option de ping -R : pour "record route"  
: chaque commutateur enregistre son adresse dans le paquet.

- utilisation pour tracer les itinéraires
- utilisation des traces d'ICMP liées aux mécanismes de routage ("redirect error").

## **Programme traceroute permettant de tracer ou de tester une route**

### **- traceroute "adresse IP"**

. Retourne pour chaque commutateur visité un paquet ICMP à l'expéditeur avec la valeur de la durée de vie TTL Time To Live.

. Permet de connaître le chemin emprunté et la durée pour atteindre chaque routeur (comme ping -R pas toujours disponible).

### **- traceroute -g "adresseIP"**

. Permet également de tester les possibilités de routage par la source (-g routage faible force IP à emprunter les routeurs spécifiés) (-G routage strict)

. Force IP à parcourir strictement un chemin)

# IGMP

## "Internet Group Management Protocol"

### Objectif

- Utiliser les capacités de transmission IP pour permettre des diffusions sur groupe.

=> Notion de routeur diffuseur.

=> Nécessité d'une **gestion dynamique de l'appartenance à un groupe: IGMP**

(la notion de groupe s'applique à la totalité du réseau internet)

### Composition des groupes.

- Les hôtes s'associent aux groupes en émettant une requête à leur routeur (diffuseur) de rattachement comportant l'identificateur du groupe et l'interface qui doit recevoir les messages.

- Un routeur diffuseur envoie une requête périodique sur toutes les interfaces où il doit délivrer des messages diffusés.

- Utilisant ces requêtes et ces réponses un routeur gère sa table locale des groupes.

**Z En IPv6 IGMP est intégré à ICMP.**

## **Conclusion Internet Protocol**

- Un protocole de niveau 3 en **expansion considérable** (due surtout à HTTP et bientôt à la téléphonie )
  
- Il couvre de très nombreux besoins en termes d'**interconnexion pour des réseaux de transmission de données informatiques.**
  
- Il **intègre toutes les nouvelles offres de moyens de communication**
  
- Il possède une version 6 récente pour supporter le développement en termes d'adressage et de hiérarchisation.

**Compte tenu de son extension et de son adéquation IP devrait continuer d'être utilisé comme protocole unificateur de niveau 3 pendant très longtemps.**



## Questions ouvertes

- Est ce que IPV6 va réussir à se déployer dans de bonnes conditions en même temps que IPv4 va régresser?

- Est ce que IP va pouvoir s'ouvrir à un ensemble plus large et plus divers d'usagers et de besoins: essentiellement pour des usages de type commerciaux?

=> Support de la qualité de service (introduction de protocoles nouveaux MPLS)

=> Dans le cadre d'un développement important.

=> Pour des applications variées multimédia.

**Comment IP et plus généralement l'Internet supportera son propre effet sur la société?**

## **Bibliographie Internet Protocol**

W.R. Stevens "**TCIP/IP Illustrated,  
The protocols**", Addison Wesley

S.A. Thomas "**IPng and the TCP/IP  
protocols**" Wiley

A.S. Tannebaum "**Computer Networks**"  
Prentice Hall 1996

Cisco "**Internetworking Technology**"  
Publication interne

E. Gressier "**Cours réseau**",  
Institut Pasteur 1994

# **Le niveau transport**

Le niveau transport

## **CHAPITRE I**

### **Problèmes généraux de réalisation des protocoles de transport**

# Plan du chapitre

## 1. Introduction

## 2. Généralités

2.1 Choix de conception du niveau transport

2.2 Particularités des problèmes de transport

2.3 Incidence sur les mécanismes de connexion et de communication

## 3. Problèmes d'adressage au niveau transport

3.1 Nature des adresses de transport

3.2 Détermination des adresses des processus communicants

## 4. Gestion des connexions

4.1 Multiplexage et éclatement

4.1.1 Multiplexage

4.1.2 Éclatement

4.2 Problèmes d'ouverture de connexion

4.3 Problèmes de libération de connexion

## 5. Problèmes de transfert de données

5.1 Contrôle d'erreur

5.2 Contrôle de flux

5.3 Segmentation

## 6. Conclusion

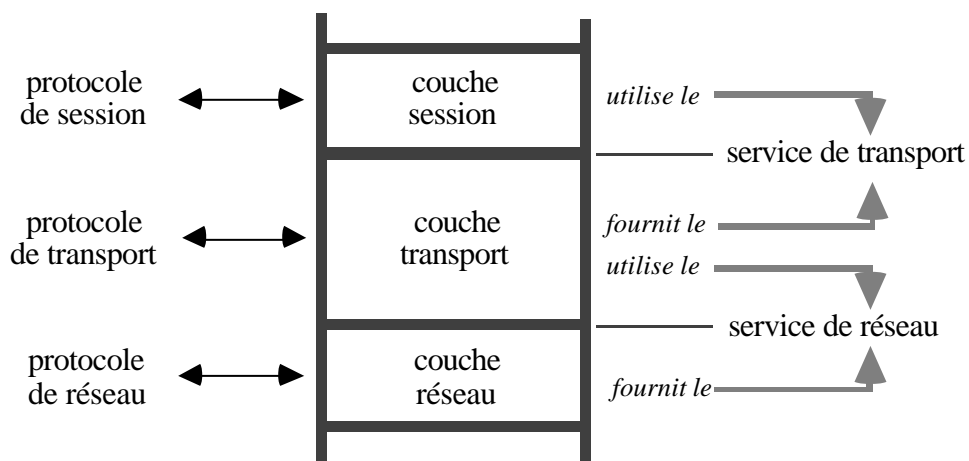
# 1 INTRODUCTION

## Objectifs du niveau transport

Offrir un service de transfert d'informations pour les besoins d'un utilisateur (en principe une entité de session).

Service de communication:

- . de **processus à processus** (de bout en bout)
- . **en point à point**,
- . **fiable**,
- . **efficace** (économiquement et en performances)
- . **indépendant** de la nature du ou des sous-réseaux traversés (LAN, MAN, WAN) .



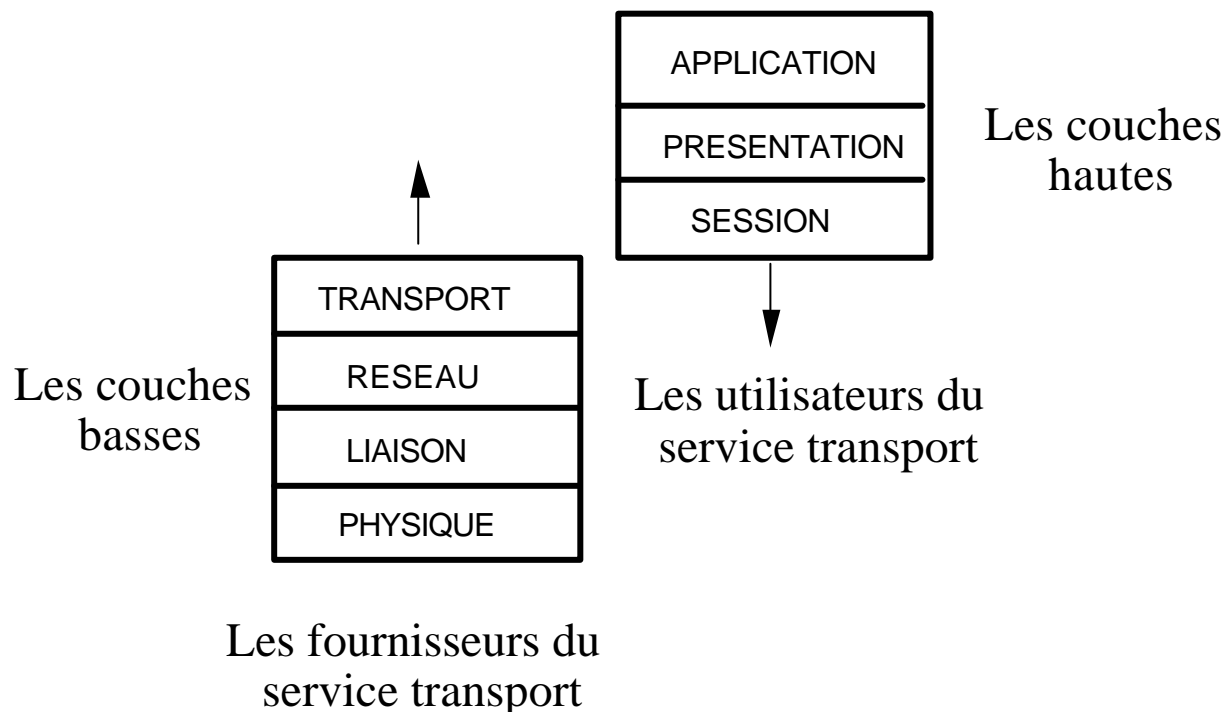
## *Importance du niveau transport*

. La couche transport n'est pas une couche de plus dans une architecture de réseau comme celle du modèle OSI.

- C'est le **coeur d'un système réparti.**

La base sur laquelle reposent les applications réparties.  
Des fonctions supplémentaires très utiles sont développées dans les couches hautes  
Une application peut se suffire du service de transport.

- C'est la **couche qui résume les couches associées à la transmission** (couches basses).



# Nécessité de la couche transport

Le niveau réseau dans les réseaux longue distance (X25, IP)

A fortiori le niveau liaison dans les réseaux locaux:

Sont **insuffisants pour les besoins applicatifs**.

## 1. Problème des erreurs de transmission

Un utilisateur de moyen de communication ne doit plus avoir à se préoccuper des erreurs de transmission.

- **Le taux d'erreur résiduel acceptable** (erreurs non détectées, non corrigées) est d'environ  $10^{-12}$  par bit pour des transmissions de données informatiques ayant une criticité normale.

- **La plupart des services de transmission** offerts par les services réseaux n'offrent pas une telle qualité.

. Niveaux liaisons sans contrôle d'erreur

Exemple : Utilisation de LLC1 sur réseau local.

. Réseaux à datagrammes non fiables

Exemple : Internet Protocol.

. Niveaux réseaux de bonne qualité : X25 PLP

Exemple : Pannes de commutateur en X25

Génération de N-RESET.

**Des architectures réseaux reportent au niveau transport le lieu unique de correction des erreurs de transmission.**



## 2. Problèmes d'adressage

- Les niveaux liaison et réseau sont des **niveaux de transmission entre entités physiques**:

Exemple: contrôleur de communication, processeurs,

- Adresse **utilisée** = Adresse d'**équipement** physique

Exemple : adresses de contrôleurs de communication série, adresses Ethernet, adresses X25, adresses internet.

- Toute solution visant à utiliser les adressages précédents pour faire communiquer des processus est plus ou moins "bricolée".

Exemple : Extension d'adressages X25 ou Ethernet

- Les **adresses utilisées pour les processus doivent être indépendantes** des réseaux utilisés.

"On ne peut pas faire dépendre la structure de l'adresse d'une application de la nature du réseau qui permet de l'atteindre".

- Au niveau transport on réalise la mise en correspondance d'une adresse "**logique d'activité**" (adresse transport) avec une "**adresse physique d'hôte**" support de l'application.

### 3 Gestion des connexions

**Les deux modes principaux des stations de transport: avec ou sans connexion.**

**. Avec connexion (TCP et ISO 8072)**

Tout échange présente nécessairement trois phases délimitées dans le temps

Établissement de connexion

Transfert de données

Fermeture de connexion

Mode le plus répandu jugé souvent nécessaire pour les applications informatiques visées.

Mode qui pose au niveau transport des problèmes d'implantation non négligeables.

Permet la gestion de spécifications de qualités de services adaptés aux besoins des applications.

**. Sans connexion (UDP et ISO 8072 ad1)**

Chaque donnée circule en contenant les informations complètes (d'adresse) lui permettant d'être acheminée à tout instant vers son destinataire.

## 2 GÉNÉRALITÉS

### 2.1 Choix de conception des protocoles de transport

*Étant donné le rôle de la couche transport et les besoins applicatifs quels sont les mécanismes de communication à intégrer dans un tel protocole ?*

**Fonctions à réaliser au niveau transport selon les options de conception (rappel)**

- **Gestion des connexions.**
  - . protocoles en mode connecté
  - . protocoles en mode non connecté.
- **Négociation de qualité de service.**
- **Multiplexage/éclatement** (des connexions de transport sur des connexions de réseaux).
- **Contrôle d'erreur.**
- **Contrôle de flux.**
- **Respect de l'ordre d'émission** (livraison en séquence).
- **Segmentation.**

# Analyse des concepteurs du transport TCP

## Un protocole unique

Adapté à IP (qui est sans contrôle d'erreur, de flux, de séquence)

=> Ces trois fonctions sont implantées au niveau transport.

Transport très optimisé et utilisable pour tous les types de réseaux.

## Conséquences

Un protocole très efficace dans son domaine

Pas de négociation de qualité de service, fonctions riches (toutes disponibles)

Mais peu adapté aux réseaux aux fonctionnalités élevées (X25).

# Analyse des concepteurs du transport OSI

- On considère **l'écart** entre le service à fournir par la couche transport et le service qu'elle obtient de la couche réseau  
=> On détermine ainsi les fonctions à mettre en oeuvre au niveau de la couche transport.
- **Meilleur est le service offert par le réseau**  
surtout en termes de contrôle d'erreur, livraison en séquence,  
**plus les fonctions** du protocole de transport **sont simplifiées.**
- Si **l'utilisateur sait quel type de niveau réseau** est utilisé (par exemple niveau en mode connecté de qualité X25, réseau sans connexion sur réseau local de type IP) **il peut sélectionner une classe** particulière de protocole adaptée à ses besoins.
- Si l'utilisateur ne sait pas si ses données sont acheminées selon une certaine qualité garantie il doit sélectionner un protocole très riche (contrôle d'erreur, de flux, de séquence).

## Conséquences

Définition d'un protocole à **plusieurs classes.**

Avec **négociation de qualité de service** (sélection de classe)

Classes aussi bien disponibles **en version simple** (pour des niveaux réseaux performants) que **compliquées.**

Protocole plus long à développer (volume de fonctions), problème d'efficacité.

## 2.2 Particularités des problèmes de réalisation du niveau transport

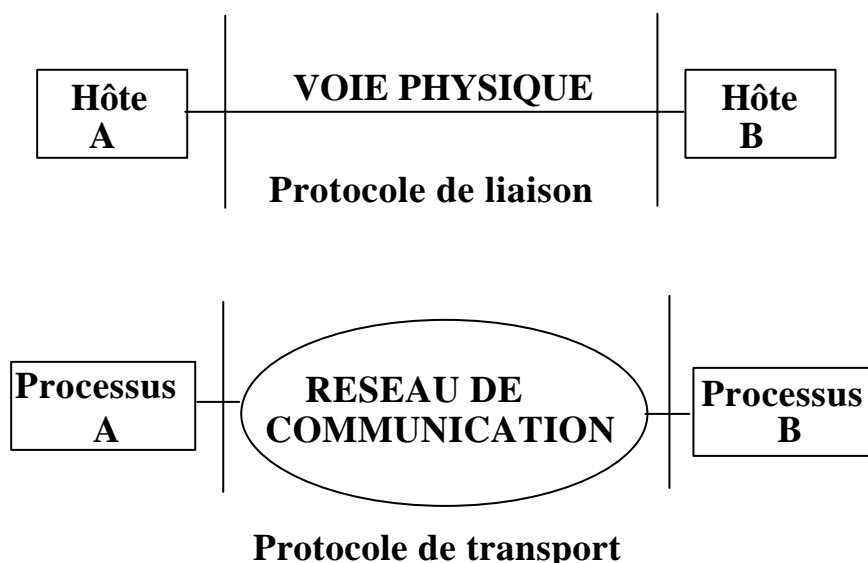
- A première vue le **service de transport** semble **voisin** du service rendu par le niveau **liaison**.

- . La communication de base est **point à point**.
- . La qualité de service attendu du transfert d'information implique dans les deux cas, **un contrôle d'erreur et un contrôle de flux**.

*Pourrait-on alors utiliser au niveau de la couche transport un protocole de liaison de données tel que HDLC ?*

Le problème principal provient de la différence entre le moyen de transmission utilisé:

- . **Une voie physique** pour le niveau liaison.
- . **Un ou des sous-réseaux** pour le niveau transport.



# Différences de comportement entre les niveaux physique et réseau dans les modes de pannes

## Modes de pannes franches

### *Niveau liaison :*

La voie de communication utilisée est le médium physique dont les modes de pannes franches sont surtout des pannes de contrôleur ou de coupure de voie.

Le logiciel de liaison est en général assez simple et bien testé. Il ne supporte que les modes de pannes du calculateur hôte.

### *Niveau transport :*

Le réseau utilisé peut avoir des modes de pannes plus variés que le niveau physique dont des pannes franches plus fréquentes.

Le processus distant est un processus usager qui supporte:

- . les différents modes de pannes logicielles d'un logiciel utilisateur,
- . les modes de pannes franche du calculateur hôte.

## Modes de pannes transitoires

### *Niveau physique et niveau réseau*

Les deux niveaux supportent les pertes de messages.

# Modes de pannes temporelles

Panne temporelle = Délai de transmission anormalement élevé par rapport aux spécifications.

## *Niveau physique :*

La capacité de stockage du canal de transmission est très faible (délai de transmission, délai de propagation) et **le temps de transmission** d'une trame est souvent **borné** et peut même être **déterministe**.

## *Niveau réseau :*

Le sous-réseau peut **stocker des paquets** pendant un certain temps et ne les **délivrer qu'après ce délai**.

- . pour des **transmissions à longue distance**,
- . en présence de problèmes de **surcharge et de congestion**.

On peut donc voir des paquets "**survivants**" arriver après un long délai (problèmes des "**vieux paquets**").

## **Hypothèses importantes de la programmation répartie**

### **Communication à délai non borné**

(Hypothèse asynchrone)

Quelle que soit **une valeur  $d_{max}$**  de délai de transmission **un paquet peut arriver après  $d_{max}$** .

### **Communication à délai borné**

(Hypothèse synchrone)

**Tous les paquets** transmis **arrivent avant un délai  $d_{max}$**  connu ou n'arrivent jamais.



## Exemples de solutions pour limiter la durée de vie des paquets à l'intérieur d'un réseau

### 1 Détermination d'une borne statique du délai de transmission en fonction de l'architecture réseau

- Utilisation d'un algorithme de routage tel que les **paquets ne peuvent boucler**.
- Majoration du délai de transmission sur le chemin le plus long en utilisant deux majorations:
  - . **majoration du délai de séjour** dans les commutateurs (destruction si l'on dépasse la valeur).
  - . **majoration du nombre de commutateurs traversés** (diamètre du réseau).

### 2 Estampillage des paquets pour la mesure du temps de transmission

- Chaque paquet transporte une information de **délai depuis son instant d'émission** ("Time to Live").
- Chaque nœud **incrémente le délai** de la durée du séjour et du temps de transmission (plus simplement on décompte le nombre de commutateurs traversés).
- Un commutateur doit **détruire** tout paquet ayant dépassé une durée de vie maximum.

### 3 Estampillage des paquets par la date de création

- Cette technique utilise une **synchronisation des horloges** des différents nœuds du sous-réseau (on doit tenir compte de l'incertitude de synchronisation).
- Lorsque le **délai maximum est dépassé** le paquet est **détruit**.

## Différences de comportement entre les niveaux physique et réseau dans les propriétés d'ordre

### Problèmes de causalité (déséquence)

#### *Niveau physique :*

- Une voie de communication physique a un comportement "**causal**": les messages émis dans un certain **ordre** arrivent dans **le même ordre**.

**Ils ne peuvent se dépasser.**

#### *Niveau réseau :*

- Le réseau (en mode datagramme) peut **déséquence** ou **dupliquer les paquets** transmis.

En raison des **algorithmes de routage** dans les réseaux à datagrammes.

En raison du **problème de délai de propagation** précédemment évoqué:

- . un paquet en délai de garde est retransmis
- . la première tentative arrive correctement longtemps après.

# Incidence sur les mécanismes de connexion et de communication au niveau transport

**Éléments des problèmes à résoudre qui vont être développés dans la suite.**

## **- Adressage**

*Niveau liaison* : Un équipement n'a pas de problèmes pour spécifier avec quel autre équipement il veut communiquer, puisqu'il n'est relié qu'à un petit nombre de sites distants voire un seul équipement.

*Niveau transport* : L'adressage des destinataires est complexe, puisqu'il y a en général de nombreux hôtes et de nombreux processus sur ces hôtes dans le réseau.

## **- Gestion des connexions**

*Niveau liaison* : Il suffit de s'adresser à l'entité située à l'autre extrémité de la ligne ; elle est toujours présente, sauf en cas de panne. La gestion des paramètres de qualité de service est simple.

*Niveau transport* : Le mécanisme d'établissement est beaucoup plus complexe en raison des problèmes de fonctionnement du niveau réseau (déséquencement et délais d'acheminement).

## - Contrôle d'erreur

*Niveau liaison*: Le mécanisme se réduit à la détection d'erreurs et à la retransmission après temporisation.

*Niveau transport* : Il s'agit de récupérer des erreurs qui n'ont pu être corrigées par les niveaux inférieurs, aussi bien les erreurs signalées que les erreurs non signalées et surtout de faire face à de "vieux" paquets.

## - Contrôle de flux et aléas des taux de transmission.

*Niveau liaison* :

On peut se contenter d'allouer **un nombre de tampons fixe** pour chaque liaison traitée. Un contrôle de flux simple basé sur une fenêtre de taille fixe est donc satisfaisant.

*Niveau transport* :

Une entité peut avoir à établir et à gérer plusieurs connexions avec une autre ou plusieurs autres entités. Les délais de traitement des opérations peuvent être très variable et très importants.

On ne peut plus faire de **l'allocation statique de tampon**. On s'oriente alors vers un contrôle de flux reposant sur **une fenêtre de taille variable**.

## 3. Problèmes d'adressage au niveau transport

### 3.1 Nature des adresses de transport

*. Adresse = nom local de processus*

Solution très médiocre pour résoudre différents problèmes comme:

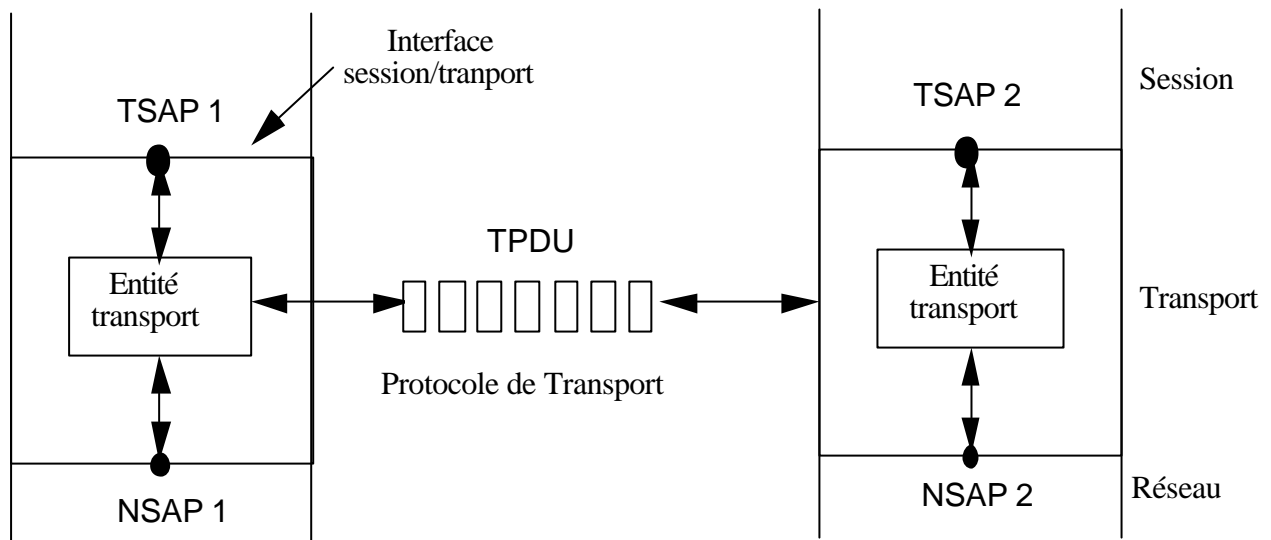
- La migration des processus.
- Le remplacement en cas de panne.
- L'existence de services pouvant être rendus par plusieurs instances d'un même code.
- Les variantes de syntaxe des noms de processus sur différentes machines.

*. Adresse = nom d'une file d'attente, d'une boîte à lettre, d'un port, d'une porte, d'un point d'accès de service transport (TSAP), d'une prise ou "socket".*

- Il y a indirection par un point de passage intermédiaire auquel se raccrochent les usagers.
- L'adresse est d'une syntaxe spécifique unique pour toutes les machines du réseau.

Exemple : TCP entier sur 16 bits.

## 3.2 Détermination des adresses des processus communicants



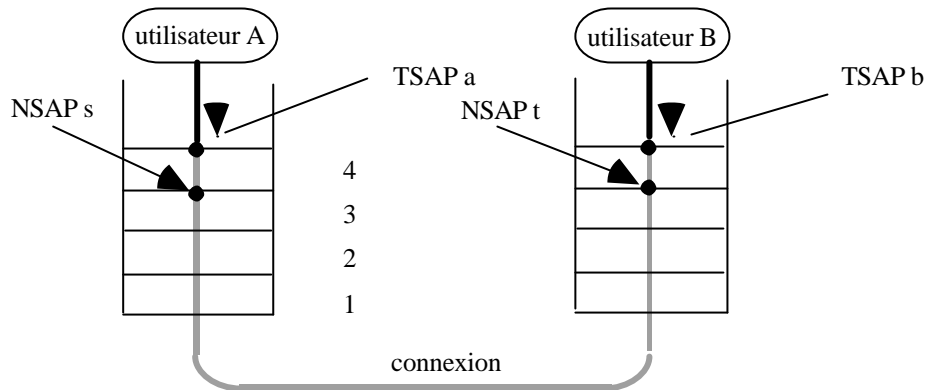
L'ouverture d'une connexion suppose la connaissance des adresses.

**TSAP : Point d'accès du service de transport** ("Transport Service Access Point") : comment sélectionner le processus qui rend le service.

**NSAP : Point d'accès du service de réseau** ("Network Service Access Point") : sur quel site se trouve ce processus.

Le routage réseau se charge ensuite de le trouver.

## Exemple d'une connexion en mode client-serveur



- 1/ L'utilisateur B du service de transport (**le serveur**) se rattache au TSAP **b** et se met en attente d'une indication de connexion entrante.
- 2/ L'utilisateur A du service de transport (**le client**) qui désire établir une connexion de transport avec B émet un T\_CONNECT.request en indiquant son adresse d'appelant (TSAP **a**) et l'adresse de l'appelé (TSAP **b**) .
- 3/ L'entité de transport locale à A doit sélectionner un NSAP sur sa machine (NSAP **s** pas difficile à obtenir) et connaître un NSAP de la machine où réside B (NSAP **t**) (c'est l'utilisateur A ou un mécanisme de localisation qui va lui fournir).

Elle demande l'établissement d'une connexion de réseau entre ces deux NSAP (N\_CONNECT).

4/ Une fois la connexion de réseau établie, l'entité de transport locale à A peut communiquer avec l'entité de transport locale à B (phase de transfert de données sur la connexion de réseau).

5/ L'entité de transport locale à B émet alors un T\_CONNECT.indication vers le TSAP m à destination de B (qui l'attendait depuis l'étape 1) ;

B peut alors répondre par un T\_CONNECT.response qui se traduit du côté de A par un T\_CONNECT.confirmation : la connexion de transport entre A et B est alors établie.

*Le scénario marche parce que B était en attente sur le TSAP b et que A est supposé connaître b ainsi que le NSAP t).*

***Comment A peut-il connaître ces adresses ?***

Comment établir une correspondance biunivoque entre b et le service offert par B.

Problème de liaison (analogue de l'édition des liens dans les programmes): solutions statiques ou dynamiques.



## **Première solution : liaison statique** **Connaissance codée dans les programmes**

Tous les utilisateurs du service offert par B au niveau transport ont tous connaissance de m de façon permanente.

Analogie avec le 18 numéro des pompiers.

- Principe simple mais limité.
- Les serveurs dont l'adresse est fixe sont nécessairement assez peu nombreux
  - => Cela revient à installer de façon définitive un logiciel sur un point de service donné,
  - => Services à caractère générique système
- Un client peut avoir à connaître une longue liste d'adresses de TSAP s'il utilise de nombreux services usagers.
- Il y a gaspillage à utiliser une adresse permanente pour un serveur peu sollicité.
- Il est donc préférable, dans la plupart des cas, d'utiliser des adresses de TSAP non permanentes.

## Seconde solution: liaison dynamique Processus créateur de serveur

### Chargement dynamique par un processus "logger"

#### Problème

*Si l'on attribue des TSAP de façon non permanente, comment A peut-il utiliser une adresse qu'il ne connaît pas a priori?*

#### Solution

- On n'attribue pas une adresse permanente à chaque serveur potentiel.
- On dispose d'un processeur serveur particulier, dont le rôle est de **créer à la demande** des instances de processus serveur => il en connaîtra donc les adresses de point d'accès de service puisqu'il les créera lui même.
- L'adresse de TSAP du créateur de serveurs ("logger") est **permanente et connue de tous** les utilisateurs (solution de liaison statique précédente).

# Protocole

- 1/ Le processus créateur ("logger") a une adresse **TSAP a** permanente et connue a priori de tous les utilisateurs du service de transport.
- 2/ Le client A se connecte au **TSAP a**, en passant par son **TSAP n**, selon le scénario décrit plus haut puis indique au serveur de création le service auquel il souhaite accéder.
- 3/ La définition du service à créer peut se faire:
  - soit sous la forme d'un nom logique alors le processeur créateur doit disposer d'une table de correspondance entre les noms de service et les images binaires
  - soit directement sous forme d'un nom de fichier contenant une image binaire disponible sur le site ou sur le réseau dont le processus serveur de création demande à un chargeur le chargement.
- 3/ Le processus "logger" alloue une adresse de TSAP libre (**TSAP b**) et l'attribue à B lors de son chargement. B se rattache au **TSAP b** au début de son exécution .
- 4/ Le processus logger communique l'adresse du **TSAP b** au client A en réponse à la demande de création et ferme la connexion de transport entre le **TSAP n** et le **TSAP a**.
- 6/ Le client A se connecte au **TSAP b** pour obtenir le service souhaité.

*Ce protocole ne marche que pour les serveurs que l'on peut créer à la demande (compilateur, date,...).*

*Ce n'est pas le cas pour un serveur qui fonctionne en permanence (exemple serveur de fichiers).*

## Troisième solution: liaison dynamique Serveur d'annuaire ou de nom

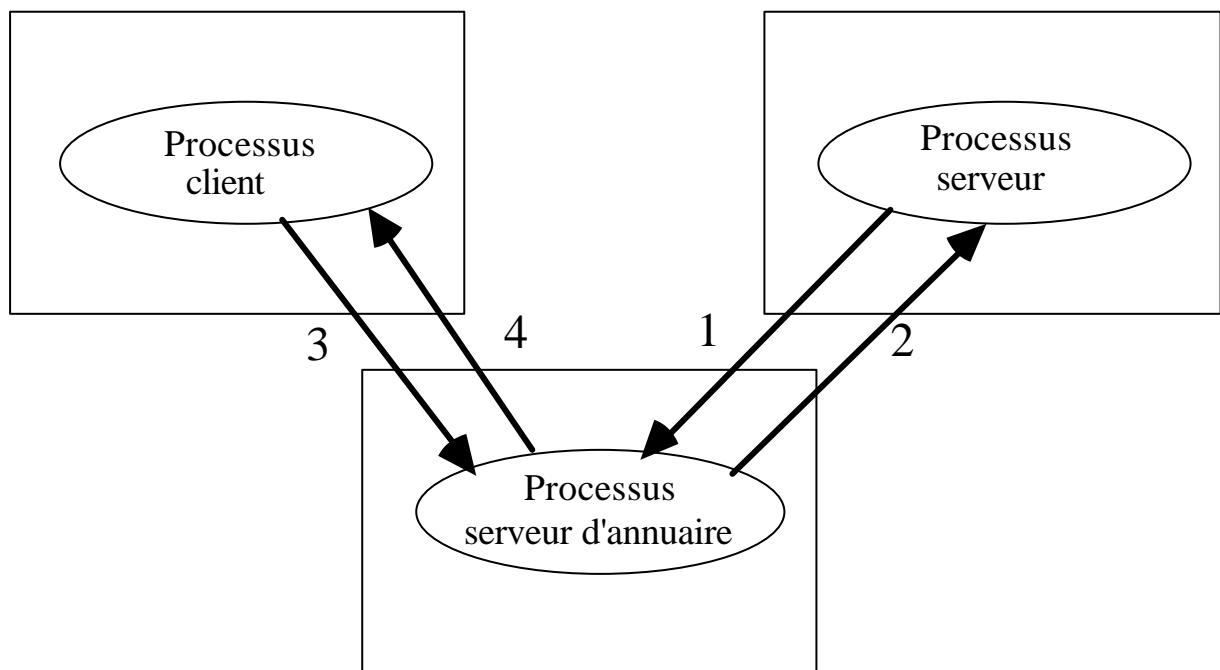
### Principe de la solution

- La solution est l'analogue de ce que réalise le 12 des renseignements téléphoniques.

- Elle consiste à faire appel à un processus **serveur d'annuaire**, dont l'adresse de TSAP est permanente et connue de tous (solution statique).

- Ce processus **gère un annuaire** ("une base de donnée") de services auprès duquel on peut s'enregistrer en tant que serveur et qui sur demande fournit des renseignements sur un service:

- principalement son **adresse** (TSAP, NSAP)
- éventuellement d'autres attributs ou fonctions (protection, description/typage du service, ...).



# Protocole

**Étapes 1, 2 : Enregistrement** dans une base de données des noms de serveurs.

1/Le serveur quelconque B établit une connexion de transport avec le processus serveur d'annuaire (tout le monde connaît son adresse).

Il demande de s'enregistrer par son **nom logique** en tant que serveur sur un TSAP b qu'il a obtenu. Il fournit:

- . le nom du service (chaîne de caractères)
- . l'adresse réseau d'accès du service (TSAP, NSAP).
- . tout attribut complémentaire nécessaire.

2/Le serveur d'annuaire **acquiesce la demande** d'enregistrement positivement (absence d'homonymie, ...).

**Étapes 3, 4: Liaison** entre un client et un serveur.

3/Le client A établit une **connexion de transport** avec le processus annuaire (il connaît aussi son adresse).

Il envoie un message spécifiant le nom logique du service dont il désire connaître l'adresse

4/ Il obtient en **réponse l'adresse du service**, A se déconnecte de l'annuaire (qui se remet en attente d'une nouvelle demande sur son propre TSAP n).

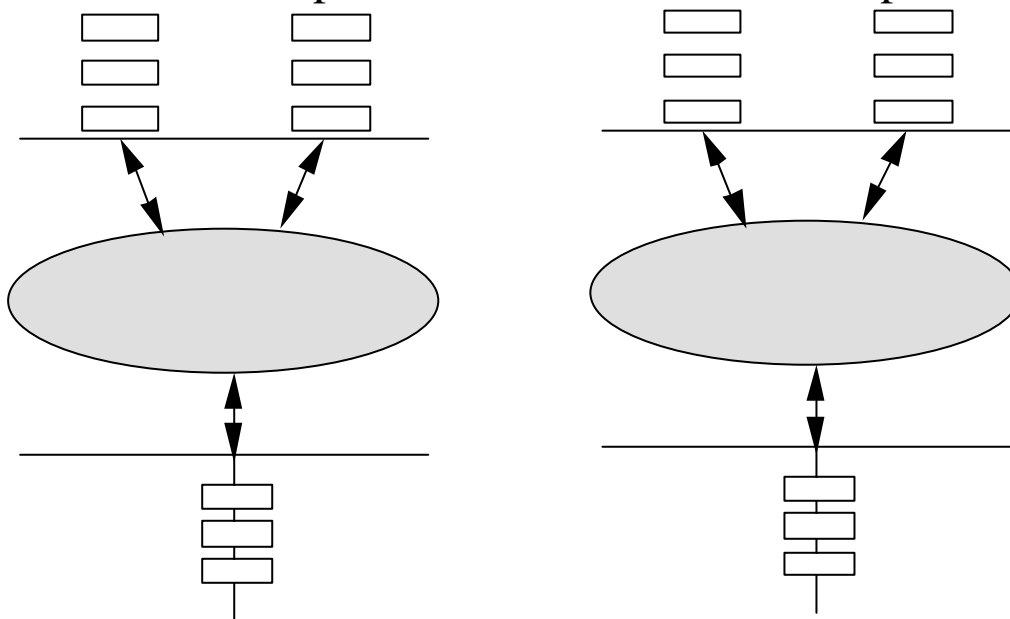
A n'a plus qu'à établir la connexion avec le serveur.

## 4. Problèmes de gestion des connexions au niveau transport

### 4.1 Le multiplexage et l'éclatement (rappel)

#### 4.1.1 Le multiplexage

Il consiste à faire prendre en charge sur une même connexion de réseau plusieurs connexions de transport.



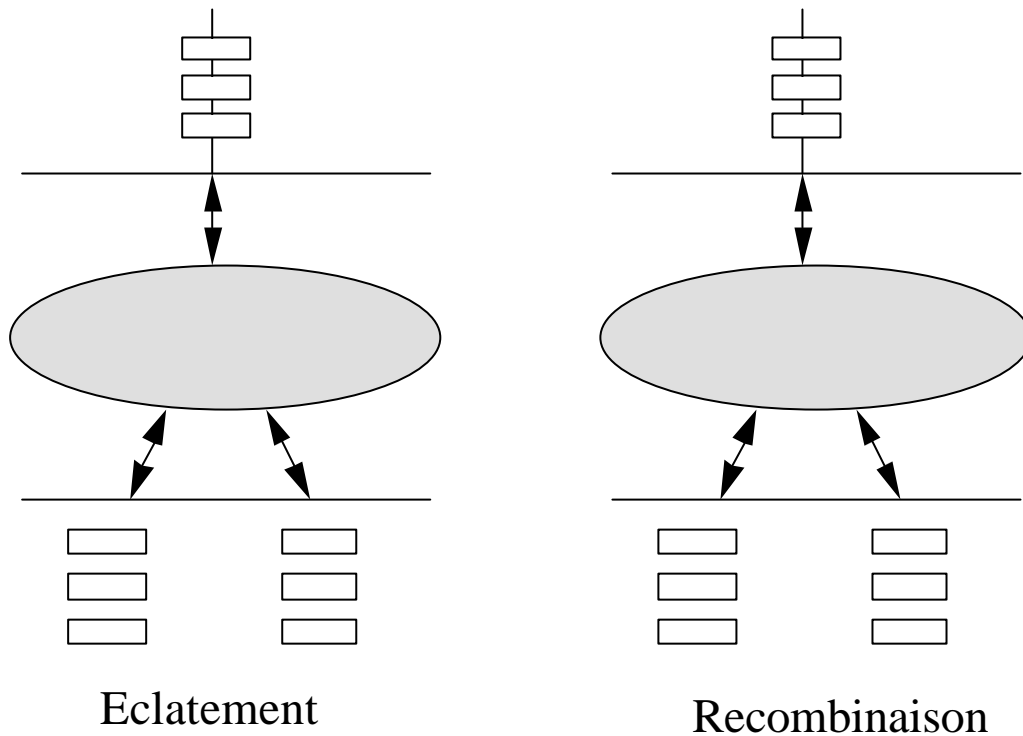
Multiplexage

Démultiplexage

**Usage** : pour optimiser les ressources de communication de la couche inférieure

## 4.1.2 L'éclatement

Il consiste à faire ouvrir à une connexion de transport plusieurs connexions de réseau, et à leur répartir le trafic.



**Usage :** Lorsque le débit des voies est insuffisant.

**Exemple d'emploi de l'éclatement:** un utilisateur important a besoin d'une connexion à haut débit de façon intermittente.

Si le sous-réseau utilise un contrôle de flux avec fenêtre et un numéro de séquence sur 3 bits, l'utilisateur doit s'arrêter d'émettre après 7 messages et attendre que les paquets soient acquittés par le récepteur (si le canal physique utilisé est un canal satellite on émet 7 paquets toutes les 540 ms).

Avec des paquets de 128 octets, le débit maximum est de 14 kb/s, même si la bande passante du canal physique est plus de 1000 fois supérieure.

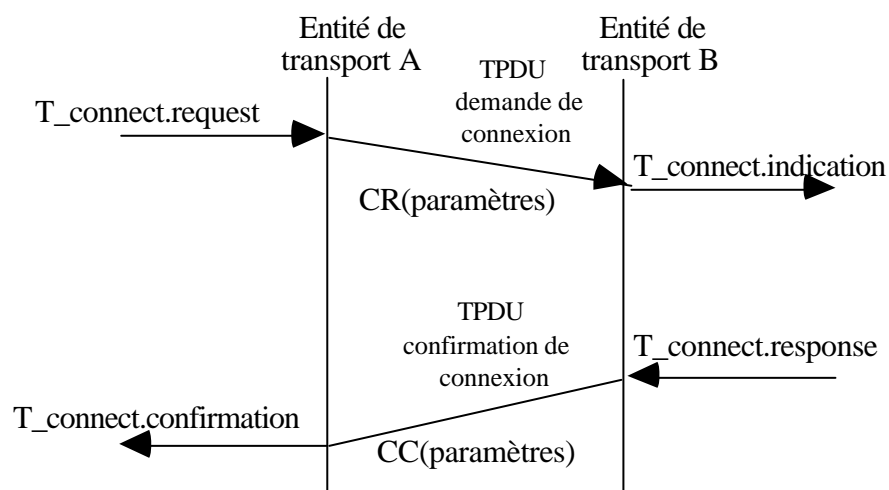
## 4.2 Problèmes d'ouverture de connexion

- L'ouverture des connexions au niveau transport est un problème difficile, car le sous-réseau peut mémoriser des paquets pendant un temps très long (éventuellement non défini dans l'hypothèse asynchrone).

Ceci est surtout vrai pour les réseaux à datagrammes.

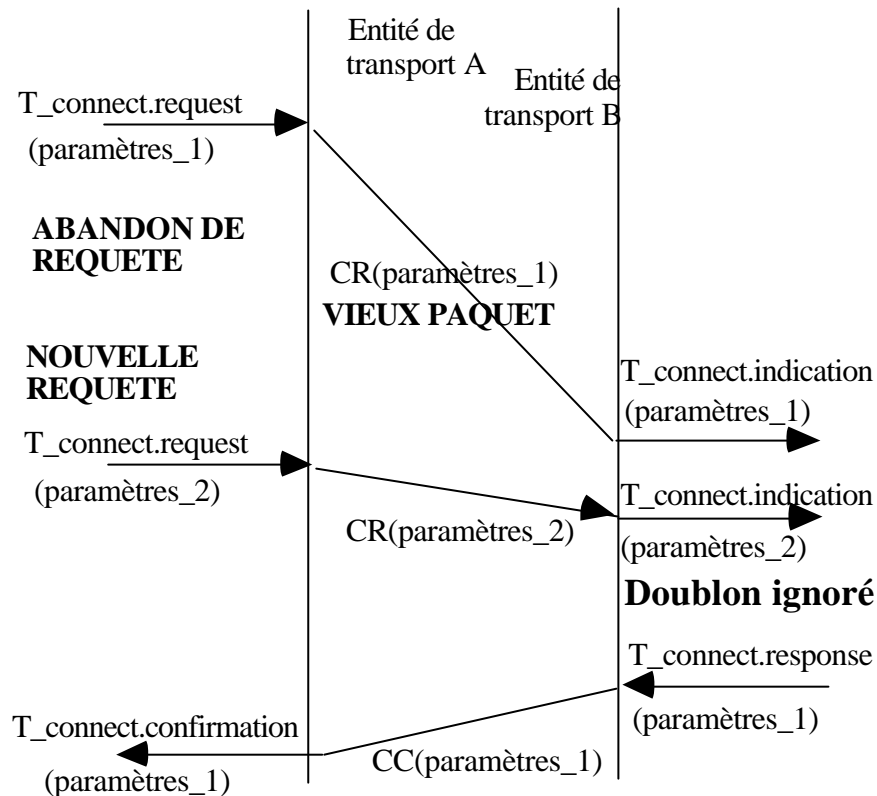
- Le schéma de base d'accord confirmé OSI devient inadapté.

Rappel du schéma de base d'accord confirmé





## Exemple 1 d'un comportement incorrect en présence d'un délai de transmission élevé.

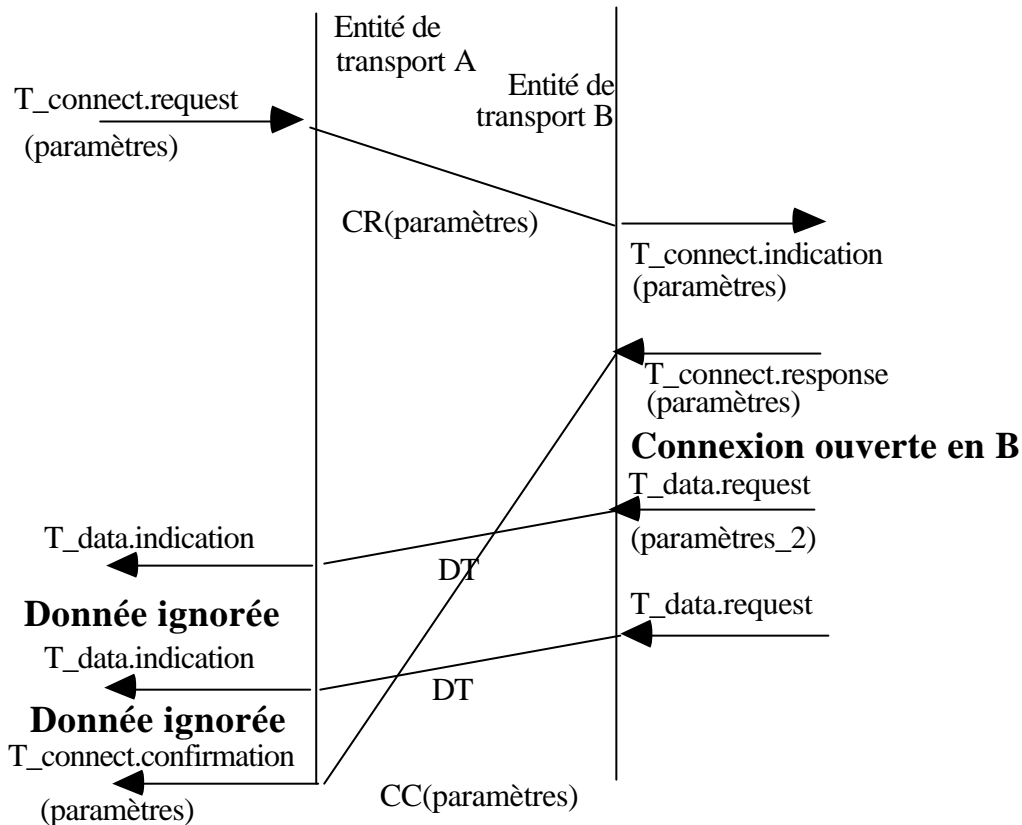


### Remarques

- Seul responsable de l'erreur le délai élevé de transmission du niveau réseau. Il n'y a pas d'erreur de transmission et la séquence des paquets est respectée.
- Sur l'exemple on voit que le problème provient surtout de l'absence d'identification des connexions.

**Solution : Notion de référence de connexion et mécanisme de gel des références.**

## Exemple 2 d'un comportement incorrect en présence d'un délai de transmission élevé.



### Remarques :

- Responsable le séquençement des paquets non respecté d'où l'existence d'une connexion considérée comme ouverte par un site B et non ouverte par l'autre A (demi-ouverte).
- La situation peut durer très longtemps si le paquet CC arrive très retardé (ou même n'arrive jamais).

**Solution : Ouverture de connexion confirmée en trois étapes ("three way handshake")**

## Références uniques de connexion

### Notion de référence de connexion

C'est un moyen de désigner une connexion de transport (a priori différent du TSAP qui désigne le point d'accès de service).

Un même usager peut gérer sur un TSAP plusieurs connexions simultanément avec un ou plusieurs usagers.

### Problème posé

Générer à la demande des références uniques dans le réseau.

### Solution 1 : Adresses de TSAP à usage unique

- On utilise une TSAP différent à chaque connexion (on s'oblige à ne gérer qu'une connexion par TSAP).
- Chaque fois qu'on a besoin d'une nouvelle référence de connexion, on crée une nouvelle adresse de transport.  
=> Il ne peut plus y avoir d'ambiguïté entre TPDU ("Transport Protocol Data Unit") circulant sur des connexions différentes ou en cours d'établissement.
- Lorsqu'une connexion est fermée, toutes les messages qui lui étaient associés sont inutilisables (sans destinataire).
- Cette solution est coûteuse et rend très difficile la liaison.

<b>Solution 2 : Référence de connexion unique (références de connexion volumineuses).</b>
---

**1) Utilisation d'une référence de connexion contenant la date de création de la connexion.**

=> Solution correcte chaque référence est différente

. Nécessite une gestion correcte des dates absolues.

**2) Utilisation séquentielle d'un espace de références si grand qu'il ne peut reboucler qu'avec une faible probabilité.**

. Nécessite d'une sauvegarde en mémoire stable du dernier numéro généré (car en cas de panne du site on ne doit pas réutiliser ceux qui lui sont inférieurs).

**3) Tirage aléatoire d'une référence dans un espace de numérotation si grand qu'on ne peut tirer deux fois successivement la même référence qu'avec une faible probabilité.**

. Nécessite d'une sauvegarde en mémoire stable du dernier numéro généré (car en cas de panne du site on ne doit pas réutiliser ceux qui lui sont inférieurs).

. Solution probabiliste (exemple sur 32 bits)

La référence unique aléatoire peut-être utilisée comme base de numérotation en séquence des messages.

### Solution 3 : Références courtes réutilisables et gel des références

. La référence doit être un identifiant **unique** composé:

- d'un entier (qui correspond par exemple à l'index d'une entrée dans la table des connexions).
- du nom du site donneur de référence

Il existe par connexion une référence pour chacune des entités communicante.

. L'entier est sur  $n$  bits,  $2^n$  est petit et il y a **réutilisation fréquente** des références par des connexions successives.

. A chaque fois qu'une connexion est ouverte on utilise une entrée libre de la table: il ne faut pas qu'il y ait d'ambiguïté entre les messages de la précédente connexion utilisant l'entrée et ceux de la nouvelle connexion.

- **Solution gel de références** : chaque référence ne peut être réutilisée pendant une période suivant la fermeture d'une connexion ce qui permet à tous les paquets en circulation sur la connexion fermée d'être éliminés.

- On doit disposer au niveau de la couche réseau, d'un mécanisme de **limitation de la durée de vie** des paquets qui permet de majorer le délai pendant lequel on risque de voir encore arriver des paquets

#### Conclusion :

Par l'un des procédés précédents on dispose d'un moyen d'identifier de façon unique les messages circulant pour une connexion.

## **Ouverture de connexion confirmée en trois étapes ("three way handshake")**

**Problème posé:** comment assurer une ouverture de connexion complète en présence d'unités de données de protocole perdues, anciennes ou déséquencées ?

**L'objectif visé:** Avant de commencer à transférer des données significatives on doit atteindre un état tel que:

- le demandeur de l'ouverture sait que le destinataire accepte l'ouverture,
- le destinataire sait que le demandeur sait qu'il accepte l'ouverture de connexion.

### **Possibilité 1**

- Ignorer les TPDU ("Transport Protocol Data Unit") de données qui arrivent avant la confirmation d'ouverture connexion.
- Ceci ne définit pas comment on atteint l'ouverture confirmée (la connexion peut rester indéfiniment en ouverture).
- Situation paradoxale puisqu'il se produit alors des pertes de données, alors que le sous-réseau est parfaitement fiable.

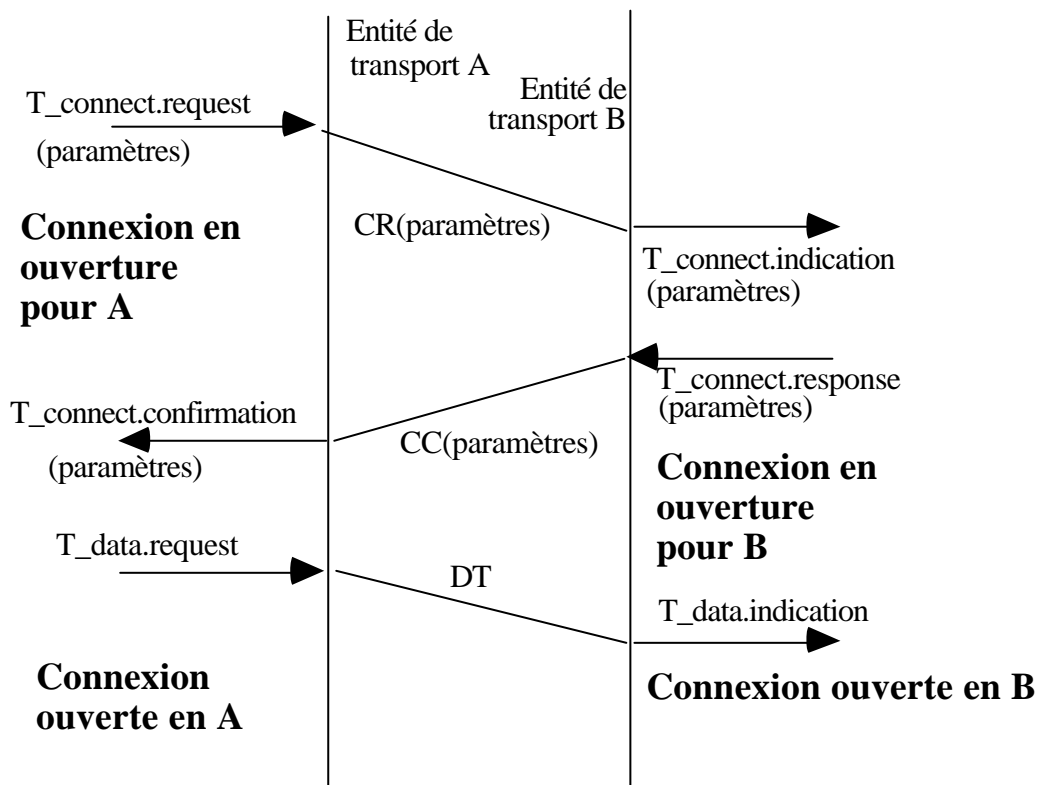
### **Possibilité 2**

- On peut stocker les TPDU de données qui arrivent avant la confirmation de connexion, afin de les remettre en séquence lorsque la connexion est établie.
- La confirmation n'est pas encore définie et on peut avoir à stocker de gros volumes de données.

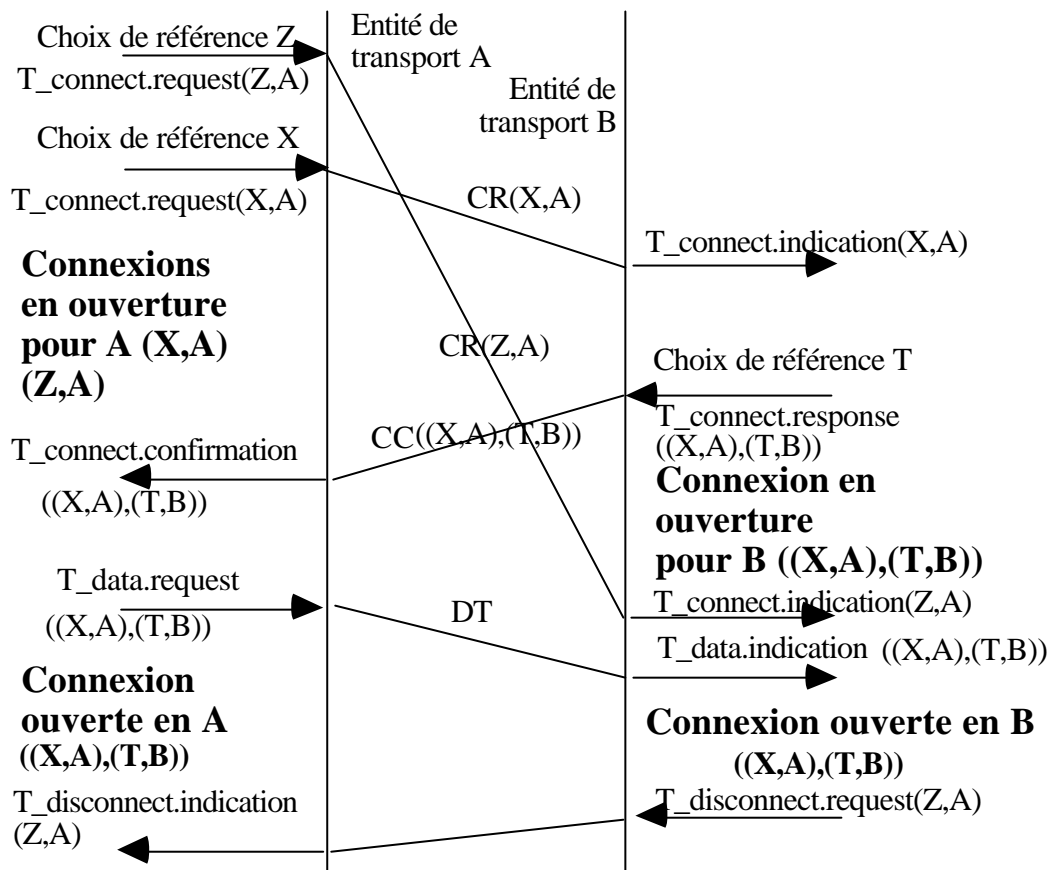
# L'ouverture confirmée en trois étapes

## Principes

- On interdit à l'entité appelée d'envoyer des messages (de considérer que la connexion est ouverte)
- Autre qu'une TPDU de confirmation de connexion
- Jusqu'à ce qu'elle ait reçu une TPDU qui lui confirme la bonne réception de sa TPDU de confirmation de connexion par l'entité appelante.
- On peut utiliser pour cela une TPDU de données si l'appelant a des données à envoyer ou une TPDU d'acquiescement.



*Exemple d'échanges pour une ouverture de connexion en trois étapes avec utilisation de références*



*Remarques:*

- Il ne peut y avoir d'ambiguïté entre les connexions référencées Z et X dont l'ouverture concurrente est en cours.
- Le site B a le choix d'accepter ou de rejeter (ici par un `disconnect.request`) l'une des deux connexions.





*Problème principal de la libération: le consensus sur la connaissance de la libération.*

A et B doivent atteindre (assez rapidement) un état où ils décident **tous les deux que la connexion est fermée**:

- Pour éviter **l'existence indéfinie de connexions** demi-ouvertes

=>L'un a décidé de fermer et l'autre ne le sait pas.

- Pour terminer les échanges dans une situation claire du point de vue applicatif.

### **Solution de base: l'accord confirmé**

Le dialogue serait alors du type :

- "J'ai fini, et toi ?"

(T\_disconnect.request,DR,T\_disconnect.indication)

-"Moi aussi, donc fin de transmission"

(T\_disconnect.response,DC,T\_disconnect.confirmation)

**Ce protocole ne marche pas en présence de tout type de délais de transmission et de pertes de messages.**

## Le problème des deux armées

La formulation **d'un problème de consensus entre deux entités communicantes** analogue au problème de déconnexion:

- Une armée A campe dans une vallée.
  
- L'armée B (son adversaire) est séparée en deux corps d'armées B1 et B2 qui occupent deux collines distinctes.
  
- A est plus forte que chacun des deux corps d'armées B, mais A est moins forte que B dans le cas où les deux corps d'armées B1 et B2 exécutent une attaque coordonnée.
  
- Le seul moyen pour coordonner l'attaque des B est l'usage de messagers qui traversent la vallée et qui sont susceptibles de mettre très longtemps à traverser en raison des ennemis, d'être capturés ...

*Sous ces hypothèses existe-t-il un protocole déterministe (qui marche dans tous les cas) en mode message asynchrone (quels que soient les pertes ou les retards) permettant aux généraux de B de gagner avec certitude (de se mettre d'accord sur l'heure d'attaque)?*

## Résultat d'impossibilité d'une solution déterministe

### Recherche d'une solution à deux messagers

- Le commandant de B1 envoie au commandant de B2 le message suivant "Attaque demain, à l'aube?",
- Le commandant de B2 reçoit le message et fait répondre "D'accord".

Ce dialogue n'est pas concluant, car le commandant de B2 ne peut pas savoir avec certitude si sa réponse a bien été reçue (non perdue ou retardée après l'attaque proposée). S'il attaque seul, il est vaincu.

### Recherche d'une solution à trois messagers

On utilise à un protocole en trois étapes, où B1 doit accuser réception de la réponse de B2 à sa proposition.

Alors, c'est le commandant de B1 qui ne peut pas savoir si sa confirmation est bien arrivée ou non.

Dans le cas de perte de la confirmation, B2 ne bougera pas, B1 attaquera seul et sera vaincu.

### Recherche d'une solution à N messagers

En utilisant à un protocole à N messagers, le même problème se pose : l'émetteur du N<sup>ième</sup> message n'est jamais capable de savoir si son message a été reçu ou non.

Pour toute solution à N messagers comme le dernier message peut ne pas arriver il ne peut être essentiel pour la solution et on doit pouvoir s'en passer.

Comme il n'existe pas de solutions à deux messagers il n'existe pas de solution (raisonnement par récurrence).

## Une solution "probabiliste et synchrone" pour la déconnexion de transport

### *Action de A*

- L'entité de transport A désirant mettre fin à une connexion envoie une TPDU de demande de **déconnexion DR** et arme un temporisateur.

- A échéance de ce dernier, si elle n'a pas reçu de réponse, elle **retransmet la TPDU**, sachant qu'on limite le nombre de retransmissions à N.

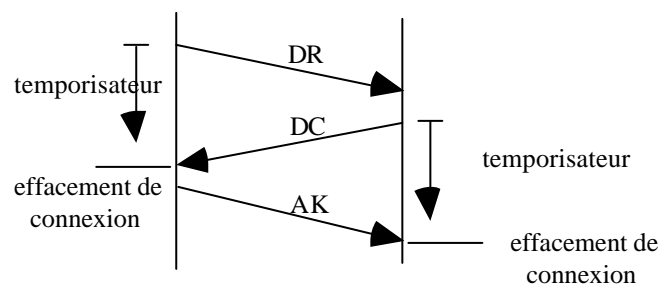
- Au bout de **N retransmissions** ou sur réception d'une confirmation, l'entité **A ferme la connexion**.

### *Action de B*

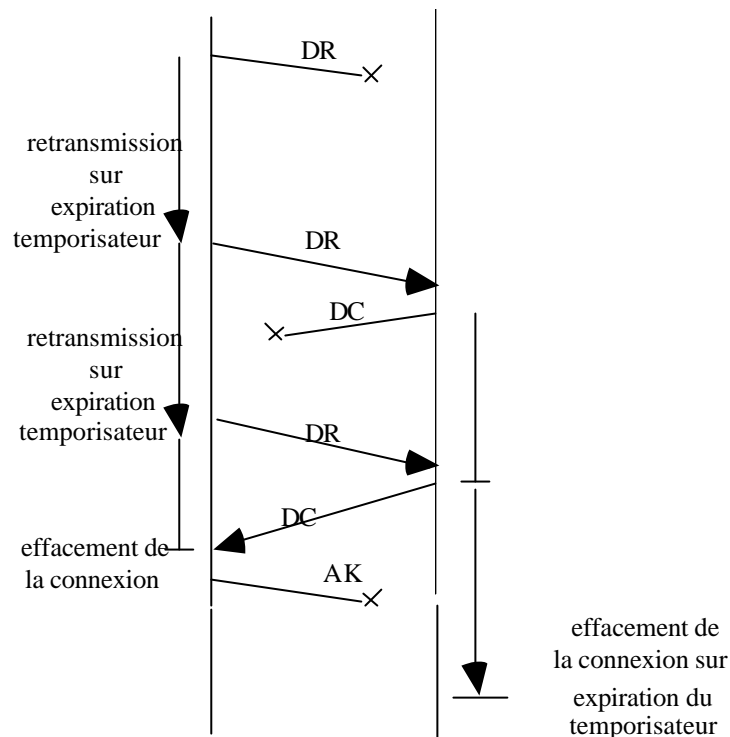
- L'entité de transport B recevant la demande de déconnexion **répond par une TPDU de confirmation de déconnexion** et arme un temporisateur.

- A **échéance** du temporisateur ou sur réception d'un **acquiescement** de sa confirmation, elle **ferme** effectivement la connexion (elle enlève de sa table des connexions ouvertes les informations concernant la connexion considérée).

- L'entité de transport A initiatrice de la libération **accuse réception de la confirmation** en envoyant une TPDU d'acquiescement et en fermant la connexion de son côté.



## Étude des modes de perte (ou de retard important) de l'une des TPDU échangées?



### Remarques

. *Les cas pour lesquels le protocole fonctionne sont basés sur l'existence de délais de garde qui permettent de régler des temporisateurs en fonction des délais de propagation des messages sur le réseau (voir le chapitre gel des références).*

. *Cas pour lequel le protocole ne fonctionne pas*

**Il existe une probabilité non nulle pour que les N demandes de déconnexion se perdent et que A ferme la connexion sans que B le sache => connexion demi-fermée (probabilité très faible).**

## **Solution synchrone (basée sur le temps) aux connexions demi-fermées**

- On utilise un **délai de détection d'inactivité** de la connexion : si aucune TPDU (de type quelconque) n'a été reçue pendant ce délai, une entité est autorisée à effacer la connexion de son côté.

- Cette solution amène à fermer des connexions qui ne devraient pas l'être uniquement **parce que les temps de transmissions sont trop élevés** par rapport aux délais de garde (à la limite aucune connexion ne peut rester ouverte).

- Pour que ce système fonctionne il faut donc que les messages aient un temps de transmission borné (**transmission à délai borné ou "synchrone"**) pour qu'on puisse régler le délai de garde d'inactivité et ne pas fermer prématurément les connexions.

### **Conclusion**

**Dans le monde des réseaux, en présence de pannes (pertes de messages, délais de transmission trop élevé), on ne construit de solutions industrielles que probabilistes et synchrones.**

# 5 Problèmes d'échange de données

## 5.1 Contrôle d'erreur

### Correction des erreurs non signalées

- Les erreurs de transmission **peuvent être corrigées** par les niveaux inférieurs (dans certaines architectures).

- Si le **contrôle d'erreur des niveaux inférieurs est inexistant** ou si les hôtes n'ont pas confiance dans le taux d'erreurs garanti par le réseau (passerelles, sous-réseaux...)

=> le niveau transport doit **calculer, transmettre et vérifier** son propre code détecteur d'erreurs.

- Un tel **contrôle de bout-en-bout** peut aussi servir à détecter les erreurs de transmission se produisant sur les lignes **hôte-ETCD réseau**, et à **déceler des anomalies de programmation** dans le sous-réseau.

- Si le sous-réseau fragmente et réassemble, **un code détecteur sur tout le message** peut aussi être utilisé pour vérifier que tous les morceaux ont été correctement transmis et remis dans le bon ordre.



## Correction des erreurs signalées

- Cas des réseaux X.25, qui indiquent une perte d'information en envoyant des paquets **RESET** ou **RESTART**.

- Dans ce cas les hôtes peuvent effectuer une reprise sur panne du sous-réseau si le protocole de transport le permet (conservation des paquets non acquittés en vue d'une reprise).

- **Exemple de fonctionnement:** après un RESET, connu de l'émetteur A, celui-ci donne au destinataire B la situation de la transmission en émettant une donnée:

"Donnée numéro 5"

=> Toutes les TPDU avant 5 ont circulé.

D'où la réponse possible en fonction de la situation de B:

"Rejet 2",

=> A doit retransmettre les TPDU (2,3,4,5).

Autre possibilité sur réception du "RESET" en B

B émet "Acquittement 2"

=> A sait que B n'a reçu correctement les données que jusqu'à la TPDU 1. Le délai de garde pour 2,3,4 retombera tôt ou tard.

## **Principes généraux du contrôle d'erreur de transport** **(très voisins de ceux du niveau liaison)**

### *Utilisation d'un code détecteur d'erreur*

Le code étant calculé le plus souvent par programme ou utilise plutôt une somme de contrôle (par exemple de type contrôle de parité) car un code polynomial est coûteux à générer.

### *Utilisation de numéros de séquence*

Le séquençement (livraison en séquence) est assuré par une numérotation des TPDU, selon un principe analogue à celui des protocoles de liaison de données.

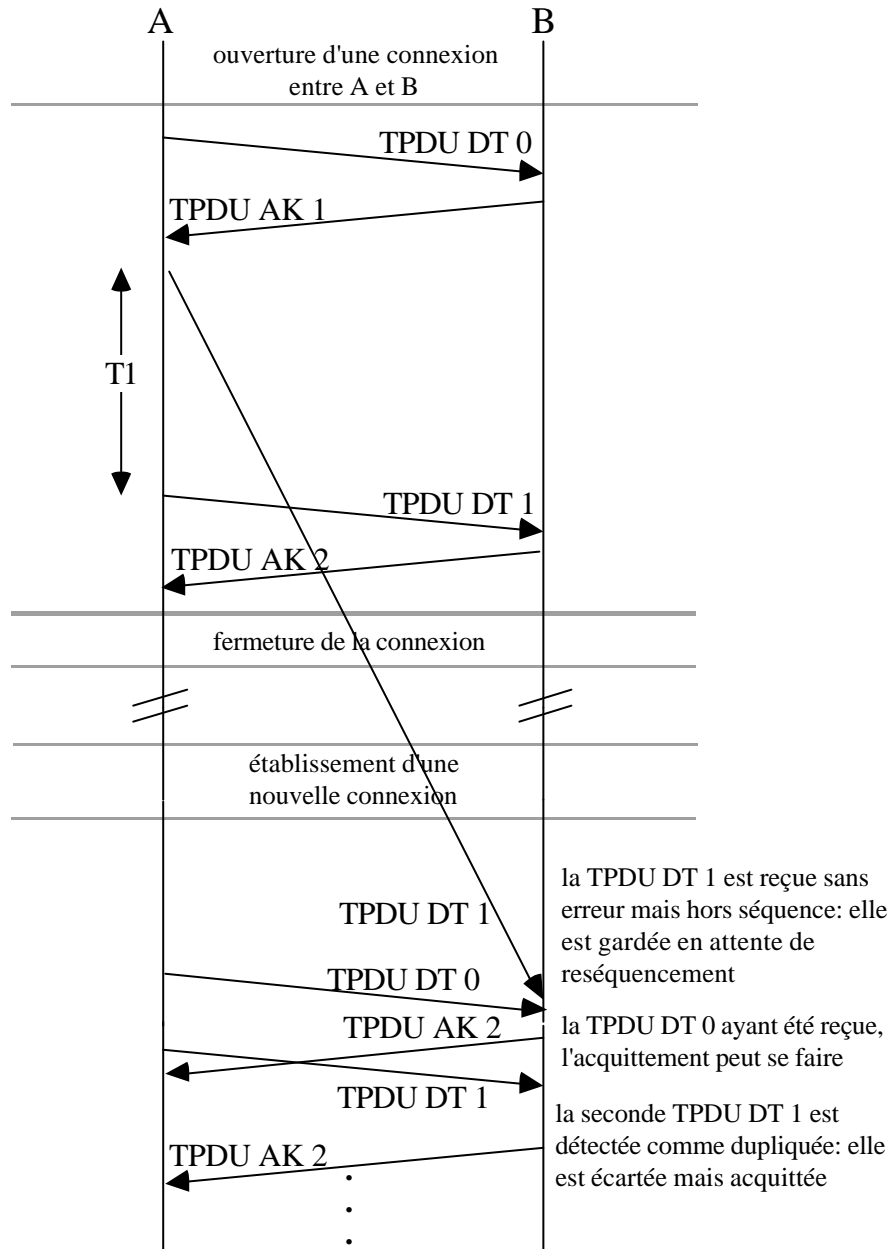
### *Utilisation d'acquitements positifs et de délais de garde*

Le contrôle d'erreur est assuré comme au niveau liaison par des protocoles à acquittement positif et retransmission sur délai de garde.

## **Difficulté du contrôle d'erreur de transport**

Il faut tenir compte des problèmes nouveaux posés par les aléas de transmission de la couche réseau.

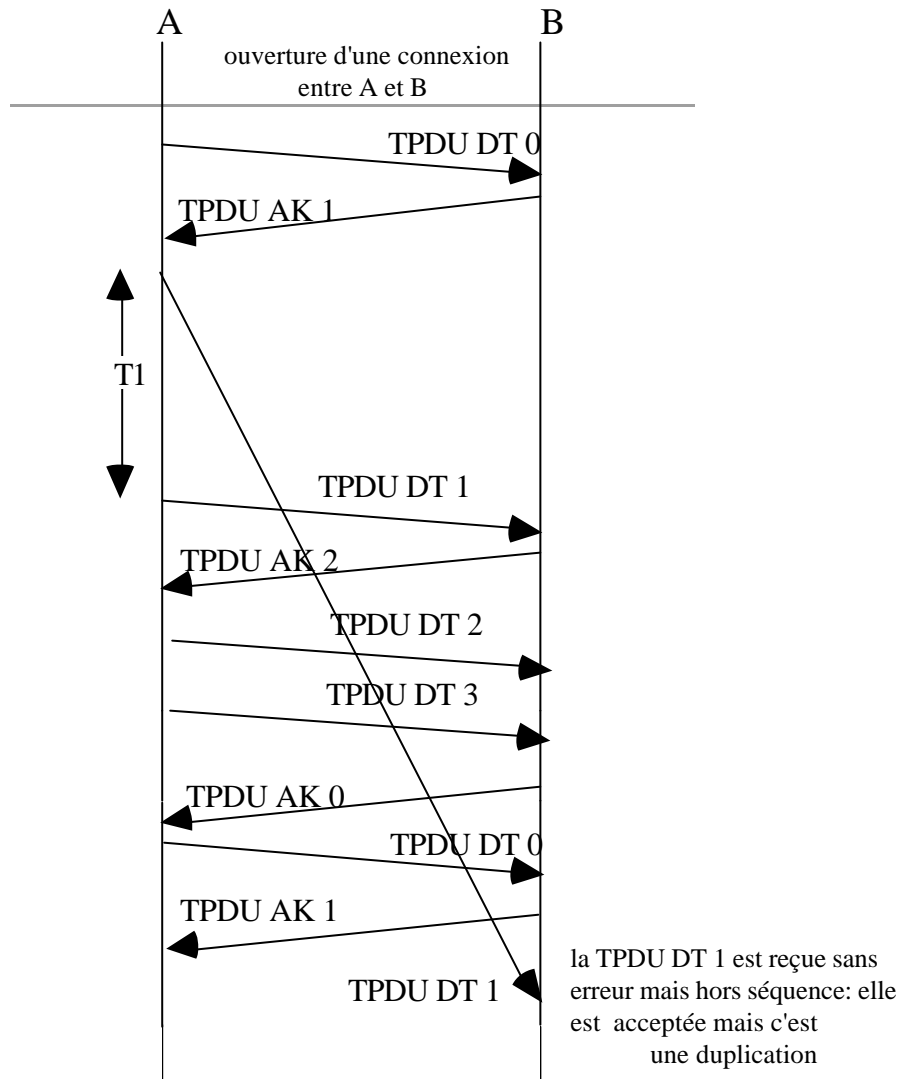
**Exemple 1 : Interférence entre paquets de données circulant sur deux connexions successives.**



**Solutions**

Utilisation de références de connexion et de gel des références. Utilisation de numéros tirés aléatoirement.

**Exemple 2 : Interférence entre paquets de données de la même connexion ayant des numéros de séquence identiques.**



**Solution**

- Utilisation de numéros de séquence de très grande amplitude (par exemple sur 32 bits) ce qui rend la probabilité de réutilisation d'un même numéro pendant une connexion infinitésimale.

## 5.2 Contrôle de flux

### Rappel

- Adaptation de la **vitesse de l'émetteur à celle du récepteur.**
- En l'absence de contrôle de flux, **des unités de données sont détruites à leur arrivée**, faute de tampons libres.
- Ce type de fonctionnement est **paradoxal dans le cas d'un niveau réseau fiable**

Il oblige **le niveau transport à retransmettre** alors que le réseau est sans erreurs.

=> Il faut donc prévoir une fonction de contrôle de flux.

## Problèmes posés par le contrôle de flux de transport

### *Nécessité d'un mécanisme de régulation très adaptatif*

- Le réseau sous-jacent **achemine plus ou moins vite** les informations selon sa charge.

- Le site récepteur permet le traitement des messages au niveau applicatif **plus ou moins rapidement selon sa charge**.

- Les **traitements d'application prennent plus ou moins de temps selon les données**.

=> Très grande **variabilité des vitesses de réception**.

=> Les commandes de ralentissement ou d'accélération du débit d'émission qui sont envoyées par le destinataire peuvent très bien donner à l'expéditeur une **image périmée** de la situation réelle.

Dans le cas où cette commande est retardée, elle peut l'amener à augmenter le débit d'émission, alors que les tampons ont été occupés entre temps.

## **Différentes solutions de contrôle de flux**

### **Solution 1 : Régulation par tout ou rien**

- L'entité de transport destinataire envoie des commandes qui autorisent ou arrêtent l'envoi d'unités de données par l'expéditeur (similaire au XON, XOFF ou au RNR, RR).
- Ce mécanisme est très mal adapté car très rigide.

### **Solution 2: Utilisation de fenêtre glissante (taille fixe)**

- Solution des protocoles type HDLC.
- Avec cette approche, le destinataire peut assurer la régulation de flux en ralentissant le rythme d'émission des acquittements, ou en l'arrêtant, ce qui provoque alors l'arrêt des expéditions des TPDU DT après épuisement d'un crédit d'émission représenté par la taille maximum de la fenêtre W.
- Cette solution peut s'appliquer mais elle ne présente pas un caractère suffisamment dynamique.

### Solution 3 : Régulation avec mécanisme de crédit

- Le principe de la régulation de flux adaptative par crédits consiste à ajouter dans certaines unités de données (principalement les acquittements AK) un champ supplémentaire qui contient un entier **CDT: le crédit**.

- Le crédit CDT représente **le nombre d'unités de données DT que l'émetteur peut émettre en anticipation** (que le destinataire est prêt à recevoir) à partir du numéro d'acquiescement indiqué.

- Le récepteur **accepte** les messages de numéro compris **numéro d'AK et numéro d'AK+CDT-1**.

- Le contrôle de flux par crédits est donc un contrôle de flux par **fenêtre glissante de taille variable** que le récepteur modifie en fonction de ses capacités de traitement (découplage total du contrôle de flux et du contrôle d'erreur).

- . le récepteur peut accroître la taille dynamiquement
- . le récepteur peut réduire la taille dynamiquement.

- Si **un acquiescement** porteur d'un crédit augmentant la taille de la fenêtre **se perd** et qu'aucun mécanisme de compensation n'est prévu **il y a interblocage**:

- . le récepteur a autorisé l'émetteur à émettre.
- . l'émetteur n'émet rien car il n'en a pas le droit.

=> Un mécanisme de **retransmission** doit être prévu.

#### Remarque

- Le crédit est donné par la **définition en absolu** des numéros de séquence autorisés (de AK à AK + CDT - 1). Une **définition relative** (autorisation d'envoi de CDT messages en plus) est impossible car les crédits doivent pouvoir être répétés.

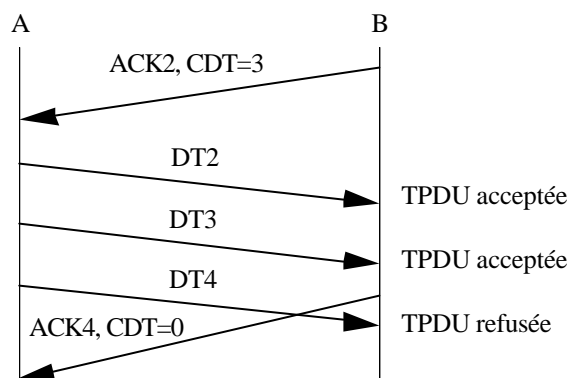


## Problème de la réduction de crédit

- Sur un réseau fiable si l'on autorise la réduction de crédit il est possible de **perdre des TPDU DT**.

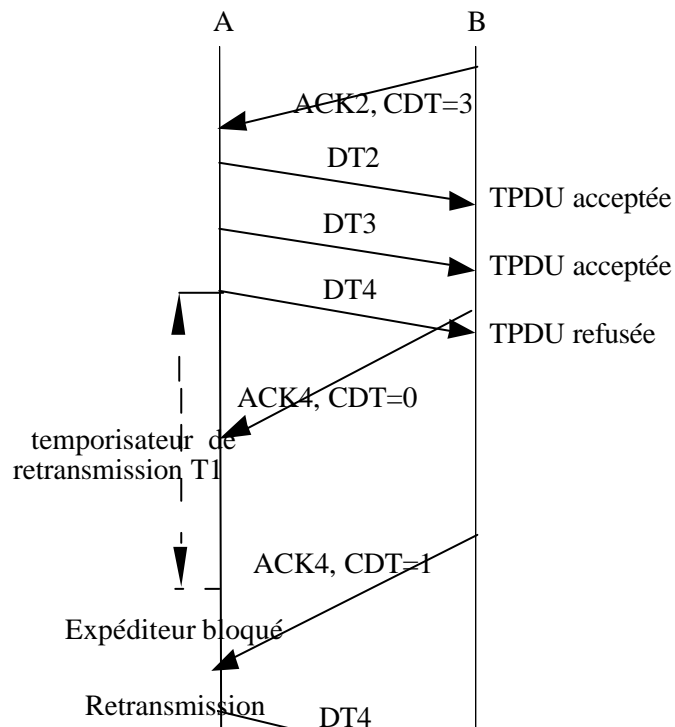
Une TPDU DT arrive hors fenêtre si le destinataire a réduit entre temps le crédit d'émission de l'expéditeur

**Exemple de perte de TPDU de données à la suite d'une réduction de la limite supérieure de fenêtre.**

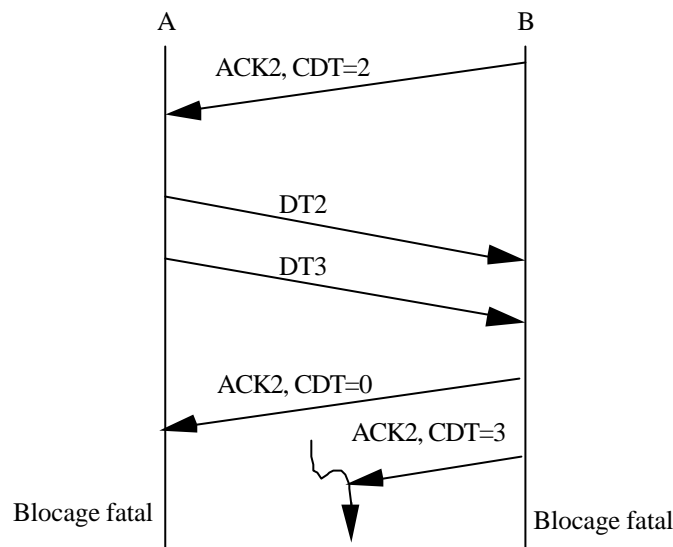


**Solution: Utilisation d'un temporisateur**

Pour tolérer les pertes en cas d'augmentation de crédit



## Problème de blocage provoqué par la perte d'un crédit (acquittement porteur de crédit)



### Solutions possibles

.1- Le **récepteur répète périodiquement** ses messages d'augmentation de crédit (constitution d'un trafic permanent de messages de type acquittement ou "Idle Data Transfer").

.2- Le récepteur **arme un délai de garde TW** après chaque message d'acquittement et répète l'acquittement à échéance uniquement si l'émetteur n'a pas repris ses émissions entre temps (il considère qu'il est perdu).

.Amélioration supplémentaire: **l'émetteur est obligé d'acquitter les messages** qui augmentent le crédit pour que le récepteur soit sûr de leur réception correcte (protocole de type PAR pour traiter les pertes de crédits).

# Discussion des solutions

## Solution 1

- Tôt ou tard le site distant recevra une copie du dernier crédit.

## Solution 2

- TW doit être réglé à une valeur élevée pour tenir compte du fait que l'expéditeur peut ne pas avoir de TPDU DT à émettre au moment de l'ouverture du crédit.

Si TW est trop faible, on risque de réexpédier des acquittements à un rythme élevé pendant un temps important, ce qui consomme inutilement les ressources du réseau.

- Par contre un délai de garde TW très élevé conduit à allonger la période de blocage de l'expéditeur quand ce dernier possède des TPDU en attente.

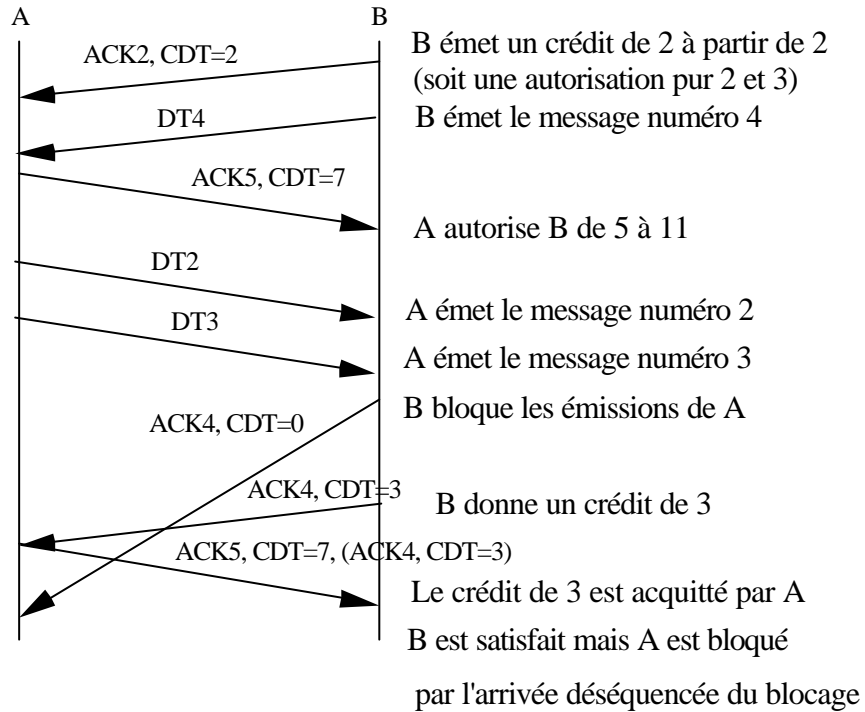
## Amélioration 2

Les crédits (portés par des acquittements) peuvent être perdus mais également déséquencés, de même que les acquittements d'acquittements.

Le déséquencement des crédits peut comme leur perte conduire à des interblocages.

Pour résoudre ce problème il faut attribuer des numéros de séquence en émission aux acquittements d'accusé de réception d'augmentation de crédits et les délivrer dans l'ordre d'émission.

## Exemple d'interblocage lié à des crédits reçus hors séquence



## 5.3 Segmentation

*Quand le réseau impose une limitation de la taille des paquets*

*=> le message doit être segmenté en paquets.*

- Dans ce cas, les données ne doivent être livrées à l'utilisateur de la couche transport qu'après l'arrivée du dernier paquets du message, ce qui oblige à marquer celui-ci par un indicateur de fin de TSDU ("Transport Service Data Unit").

- Si le réseau n'assure pas le séquençement des fragments un numéro de séquence de fragment a l'intérieur du message doit être utilisé.

## 6 Conclusion: Problèmes de conception des protocoles de transport

On dispose dans les couches transport industrielles de solutions efficaces pour résoudre de nombreux problèmes de la communication entre applications informatiques en mode message point à point.

De nombreux autres problèmes ne sont cependant pas traités

. Soit parce qu'ils sont reportés dans d'autres niveaux.

- synchronisation entre les entités communicantes
- tolérance aux pannes franches des applications.
- sécurité (au sens des violations de protection)
- ....

. Soit parcequ'ils n'ont pas encore été normalisés ou implantés efficacement dans des solutions industrielles de grande diffusion.

- communications avec délai de délivrance garanti
- communication en diffusion (multipoint)
- transport de données multimédia (protocole de transport acceptant des données informatiques, image, son). avec un niveau de qualité correspondant aux types de données.

# **Le niveau transport**

## **CHAPITRE**

### **Exemples de protocole de transport**

#### **Les protocoles de transport**

##### **INTERNET**

**UDP "User Datagram Protocol"**

**TCP "Transmission Control Protocol"**

# Plan du chapitre

## Généralités

1. UDP: User datagram Protocol
2. TCP: Transmission Control Protocol
  - 2.1 Choix de conception
  - 2.2 Format du message
  - 2.3 Le protocole d'ouverture et de fermeture de connexion
  - 2.4 La transmission des données
3. Un service pour TCP et UDP: les sockets
  - 3.1 Généralités, choix de conception
  - 3.2 Les primitives de l'interface

## Conclusion



## **Généralités**

Développés conjointement à IP  
pour former la suite TCP-UDP/IP.

### **UDP** **("User Datagram Protocol")**

Offre un accès de niveau transport à IP.

- . Protocole sans connexion
- . Protocole de transport non fiable

RFC 768 UDP Jon Postel 1980

### **TCP** **("Transmission Control Protocol")**

Un protocole de niveau transport avec une  
très bonne qualité de service.

- . Protocole en mode connecté
- . Protocole de transport fiable

RFC 793 TCP Version de base

Jon Postel sept 1981

RFC 1122 Correction d'erreurs

RFC 1323 Extensions

**1**

**UDP**

***"User Datagram Protocol"***

## Choix de conception UDP

*UDP permet l'émission de datagrammes IP en utilisant un protocole très simplifié de niveau transport.*

- **Sans connexion**  
pour des relations courtes  
simplifiant le développement du code
- **Avec adressage de transport**
- **Efficace** (en termes de performances)

*UDP implante très peu de mécanismes*

- Identification des "**ports**"  
(points d'accès de service de transport)
- Somme de contrôle d'entête  
(optionnelle)
- Aucun contrôle de séquence, contrôle d'erreur, ni contrôle de flux.

## Format du message UDP

### Entête de 8 octets.

<div style="display: flex; justify-content: space-between; padding: 0 10px;"> <span>0</span> <span>15 16</span> </div> <p>Adresse port source "Source port number"</p>	<div style="display: flex; justify-content: space-between; padding: 0 10px;"> <span>16</span> <span>31</span> </div> <p>Adresse port destination "Destination port number"</p>
<p>Longueur "Length"</p>	<p>Somme de contrôle "Checksum"</p>
<p>Zone données "Data"</p>	

### Exemple d'encapsulation UDP sur Ethernet.

Entête Ethernet	Entête LLC	Entête IP	Entête UDP	Données UDP	Postface Ethernet
14 octets	8 octets	20 octets	8 octets	<1464 octets	4 octets

## Compléments: message UDP

### **Zone source port et destination port**

Numéros de port identifiant l'utilisateur source et l'utilisateur destinataire (16 bits).

Le numéro de port source est optionnel (si non renseigné il est mis à zéro).

### **Zone longueur**

Longueur totale en octets du message.

Redondant avec la longueur de paquet IP.

=> La longueur est **optionnelle**

(dans ce cas on la met à zéro).

### **Zone somme de contrôle**

Le champ "checksum" couvre la partie entête et la partie données.

Il est calculé comme pour l'entête IP par un ou exclusif sur des groupes de 16 bits et complémenté à un.

Si la longueur des données est impaire un remplissage par des zéros est effectué sur le dernier octet.

=> La somme de contrôle est **optionnelle**

(dans ce cas on la met à zéro).

**TCP**  
*"Transmission Control Protocol"*

## Choix de conception TCP

- TCP a été conçu en fonction de IP qui est de **qualité de service médiocre**.

(type C pour l'OSI)

Le réseau sous-jacent (IP) peut **perdre, altérer, dupliquer, déséquence** les paquets.

- TCP est destiné à permettre un service de transport de données:

- . En mode **connecté**.
  - . Avec **contrôle d'erreur** (fiable).
  - . Avec **contrôle de séquence**.
  - . Avec **contrôle de flux**.
- Existence de trois phases principales
- . **Ouverture** de connexion
  - . **Transfert** de données
  - . **Fermeture** de connexion

## Format du message TCP "TCP Segment"

Un seul format de TPDU.

Pour tous les mécanismes du protocole.

=> un en-tête relativement long  
au minimum de 20 octets

	0			31					
Port source		Port destination							
Numéro de séquence									
4		Numéro d'acquittement							
Longueur entête	6	U R G	A C K	P R S H	R S T	S Y N	F I N	Taille fenêtr	
Somme de contrôle				Pointeur urgent					
Options									
Données									



## Détails concernant les différents champs

- **Numéro de port source** ("Source port")  
Entier sur 16 bits: identifie le port émetteur.
- **Numéro de port destination** ("Destination port") Entier sur 16 bits: identifie le port récepteur.
- **Numéro de séquence**  
("Sequence Number") sur 32 bits  
Si le bit SYN est non positionné c'est le numéro de séquence du premier octet de données dans la TPDU  
Si le bit SYN est positionné c'est le numéro de séquence initial (ISN): le premier octet de données a pour numéro ISN+1.
- **Numéro d'acquittement** sur 32 bits.  
Le numéro de séquence de l'octet que l'entité de transport s'attend à recevoir.
- **Longueur de l'en-tête** sur 4 bits  
En nombre de mots de 32 bits, nécessaire puisque l'en-tête comporte un champ d'options de longueur variable.
- **Champ réservé** de 6 bits.

## Détails concernant les différents champs (suite)

- Six **drapeaux** (1 bit) déterminent en fait le type de TPDU (données, acquit, libération ou établissement de connexion, ... ).

. **URG** ("Urgent") est positionné à 1 si le champ "pointeur urgent" est utilisé.

. **ACK** ("Acknowledgment") vaut 1 si le champ "numéro d'acquittement" est présent. Il sert aussi à l'établissement de connexion.

. **PSH** (Push) à 1 demande la transmission au destinataire des données du segment et de tout ce qui était en instance.

. **RST** ("Reset") ferme abruptement la connexion.

. **SYN** ("Synchronize") synchronise les numéros de séquence à l'établissement des connexions.

. **FIN** ("Fin") l'émetteur de la TPDU n'a plus de données et libère la connexion.

## Détails concernant les différents champs (suite)

- **Taille Fenêtre** ('Window Size') (16 bits)  
Représente un crédit exprimé en octets.  
Le destinataire de la TPDU est autorisé à envoyer le nombre d'octets mentionné en crédit à partir de l'octet correspondant à celui du champ numéro d'acquiescement.
  
- **Somme de contrôle** ('Checksum') (16 bits)  
Sert à la détection d'erreurs.
  
- **Pointeur urgent** ('Urgent pointer') (16 bits)  
Le drapeau URG indique au destinataire l'existence de données urgentes.  
Le pointeur urgent pointe sur la fin des données urgentes envoyées à partir du début du segment (suivi éventuellement par des données normales dans le segment).
  
- **Options**  
Permet l'échange d'informations en extension de la zone des informations  
Exemple: Option MSS ('Message Size')  
: taille maximum des TPDU.

## Protocole d'ouverture de connexion

- TCP utilise une ouverture de connexion en trois messages "**three way handshake**" pour des raisons de fiabilité.

- TCP ignore les **requêtes** de connexion **ultérieures** une fois une connexion établie entre deux points d'accès ("sockets").

- Chaque extrémité doit choisir un **numéro de séquence** (seq) sur 32 bits qui est envoyé à l'autre extrémité

Ce couple sert de **référence initiale** à la connexion (x,y).

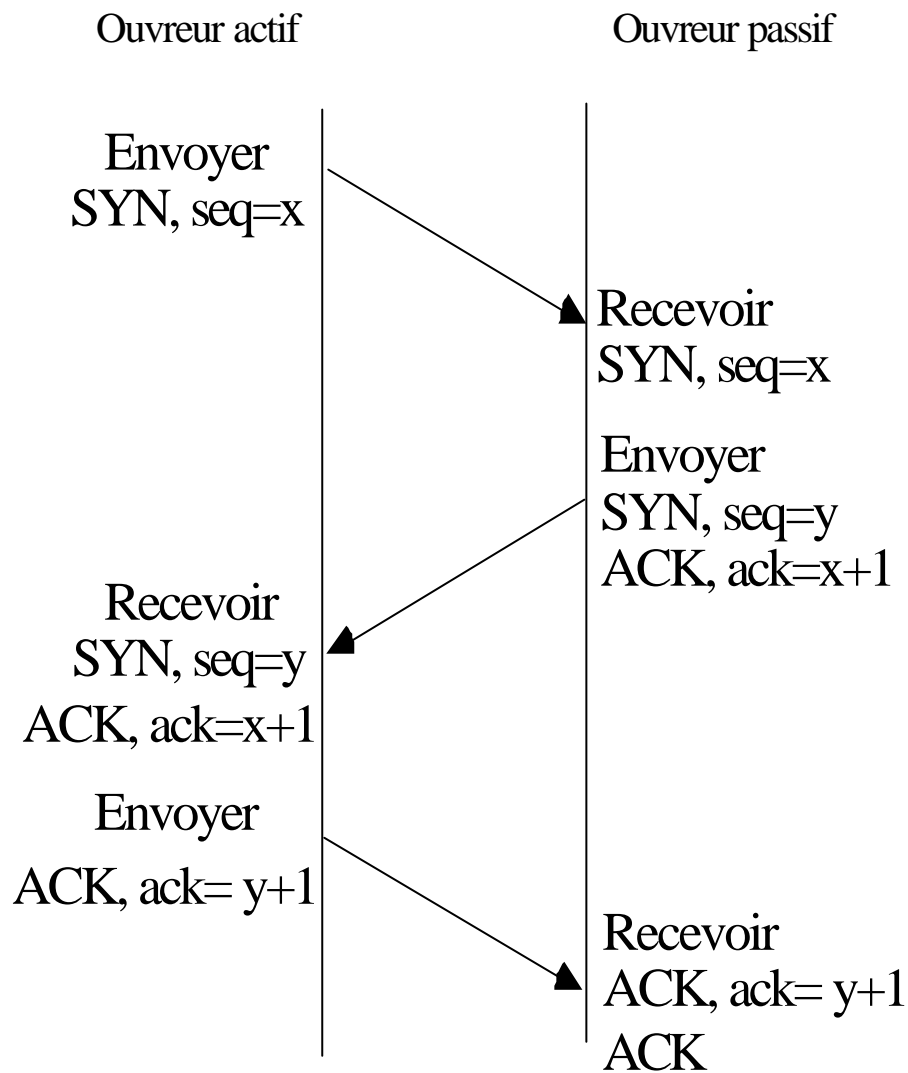
- Chaque numéro de séquence initial choisi est **acquitté**.

- La norme recommande de choisir les numéros de séquence "au hasard" **selon les bits de faible poids d'une horloge** et de ne pas réutiliser ces numéros avant un délai.

Cette technique est employée pour traiter le cas des "vieux paquets" retardés dans le réseau et qui pourraient être mal interprétés.

# L'ouverture de connexion en trois messages de TCP

## "Three way handshake"



## Les trois messages

### Message 1

- Demande de connexion avec  $SYN=1$ , un numéro de séquence initial  $X$ , et  $ACK=0$  indiquant que le champ d'acquittement n'est pas utilisé.

### Message 2

- Confirmation de connexion avec  $SYN=1$ , un numéro de séquence initial  $Y$  et  $ACK=1$  avec le numéro  $X+1$  acquittant le numéro proposé  $X$ .

### Message 3

- Acquittement du numéro  $Y$  proposé avec  $ACK=1$  et un numéro d'acquittement  $Y+1$ .

## **Approfondissement: Ouverture de connexion simultanées**

- L'ouverture de connexion distingue souvent le demandeur et l'accepteur (ici nommés actif et passif ou encore client et serveur)

Le premier site qui émet un SYN est baptisé initiateur actif ("active open")

Celui qui répond par le second SYN est l'initiateur passif ("passive open").

- Une connexion en mode bidirectionnel peut être établie selon deux modes:

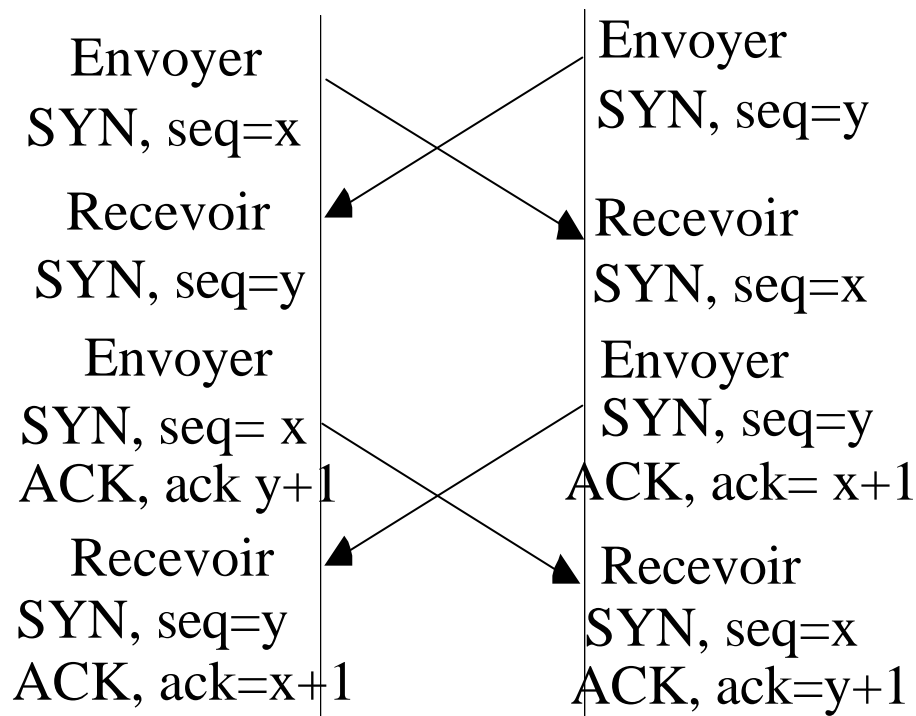
### **Cas classique**

- . Un Initiateur actif, un initiateur passif

### **Cas d'ouverture simultanée**

- . Deux initiateurs simultanément actifs

## Exemple d'échange pour une connexion avec ouverture simultanée "Simultaneous open"



### Remarque

- En cas de collision d'appel la norme indique qu'une seule connexion doit-être ouverte.

- C'est le cas ici puisque bien que les deux sites ont été actifs ils ont négociés en 4 messages une ouverture de connexion de référence unique (x,y).



## Approfondissement: Choix des numéros de séquence

- Des connexions successives doivent recevoir des numéros de **séquence initiaux différents et "aléatoires"** (de manière à différencier les vieux messages ayant circulé sur une connexion et les nouveaux).
  
- Pour des **numéros de séquence sur 32 bits** le RFC 793 TCP suggère d'utiliser pour cela une **horloge sur 32 bits**.

En fait c'est un compteur incrémenté toutes les 4 micro-secondes ce qui le fait recycler environ en 4 heures ("wrap around").

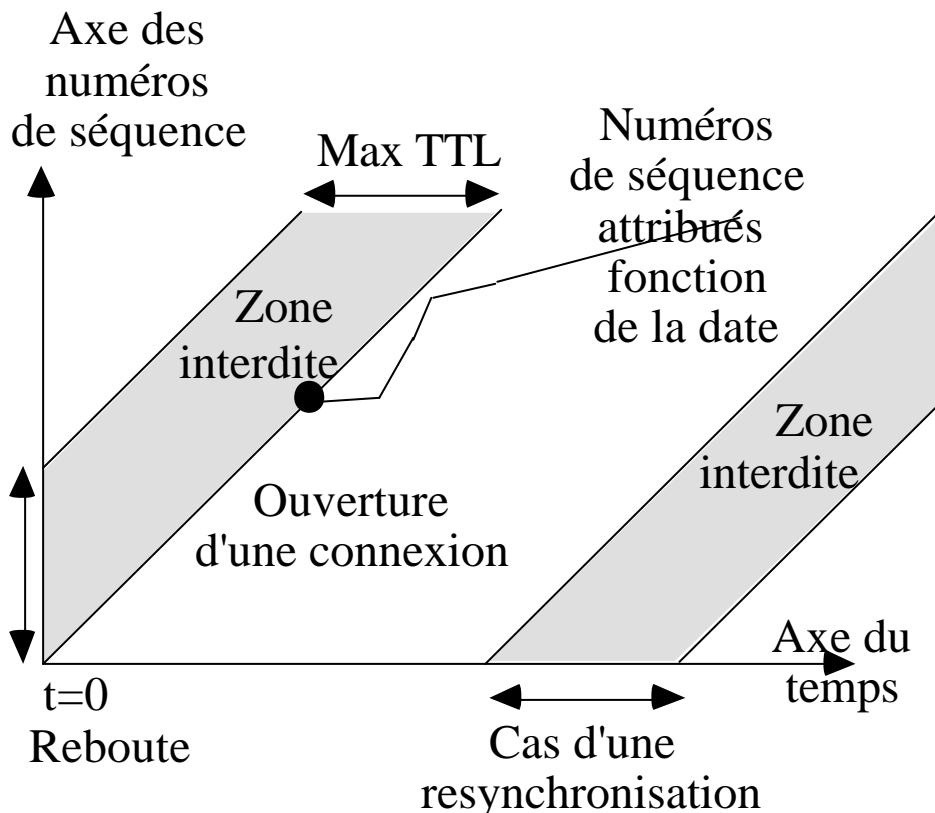
L'incrémentation n'est pas faite un par un (trop d'interruptions) mais selon une période qui peut être assez longue (exemple 1/2 seconde => incrémentation de 128000).

Remarque:

Les implantations diffèrent de façon significative sur le choix de ces valeurs.

## Choix des numéros de séquence: rappel des règles d'utilisation

- Deux numéros de séquence identiques correspondant à des messages différents de connexions différentes successives entre les mêmes ports ne doivent pas être attribués.
- On utilise la durée maximum de vie (TTL "Time To Live") d'un paquet dans le réseau de communication.
- L'usage des numéros de séquence dans la zone interdite est proscrit car des paquets ayant ces numéros peuvent être en transit.



## **Protocole de fermeture de connexion**

- La libération normale de connexion peut se faire de deux façons.

### **Libération abrupte (TCP Connection reset)**

- Lorsqu'un segment est transmis avec le bit RST positionné une libération inconditionnelle de la connexion est réalisée:

Le récepteur abandonne immédiatement la connexion.

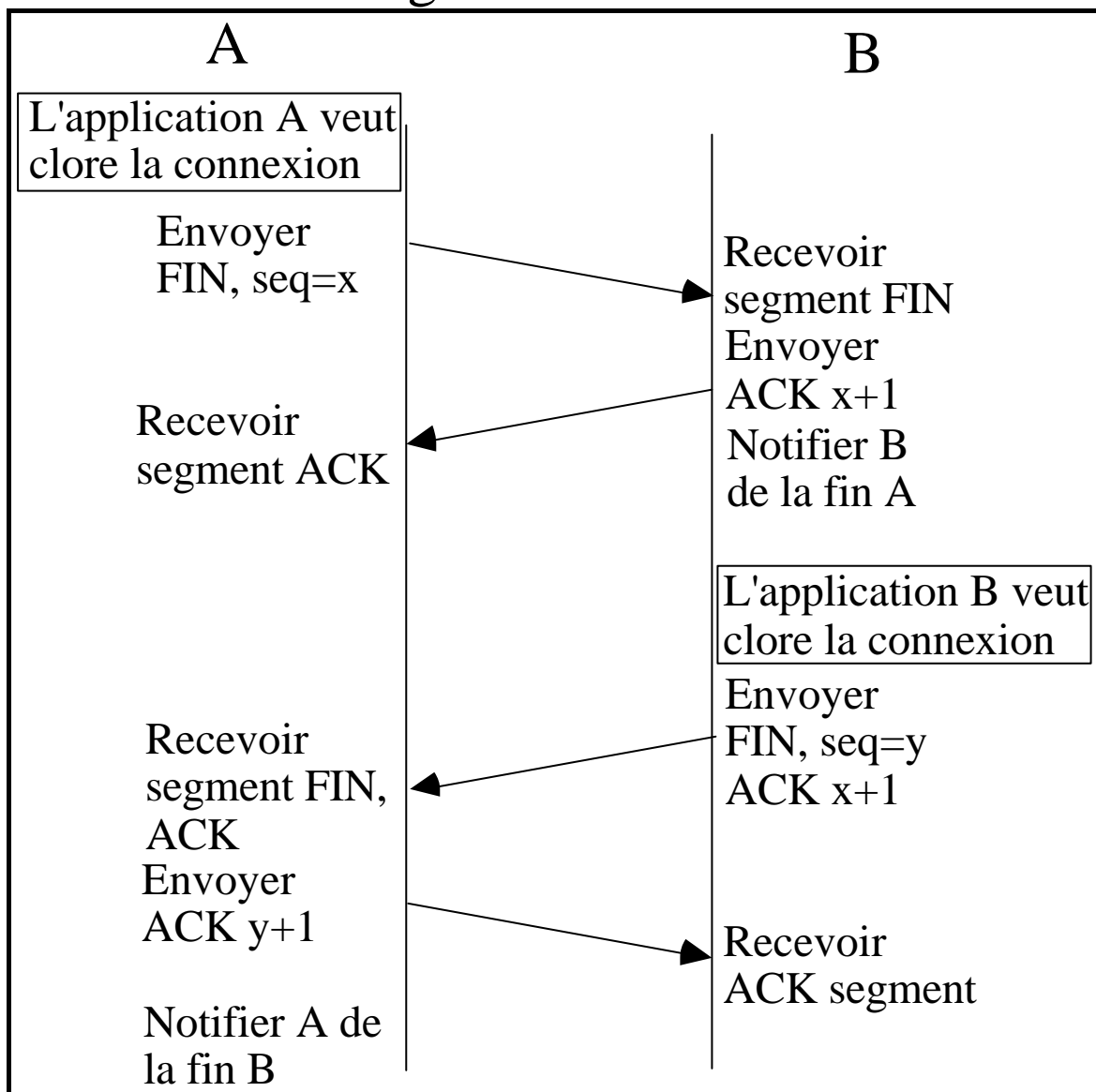
Il notifie à l'application la fin immédiate de la connexion

Il y a libération de tout l'espace de travail occupé par les messages en cours de transmission.

## Libération ordonnée (graceful close).

- Chaque utilisateur impliqué dans la connexion doit générer sa propre requête de déconnexion.

- On garantit la livraison correcte de toutes les données échangées sur la connexion.





## **Phase de transfert des données**

- En TCP les données échangées constituent un flot d'octets ("stream").

- Les données sont envoyées message par message (segment par segment en TCP).

TCP décide du moment où envoyer un segment et de celui où délivrer des données reçues à l'utilisateur destinataire.

## **Contrôle de séquence**

- Chaque octet est numéroté sur 32 bits (avec une numérotation modulo  $2^{31}$ ).

- Chaque segment contient le numéro de séquence du premier octet du champ de données.

## **Contrôle de flux**

- Le contrôle de flux utilise un mécanisme de crédit s'exprimant en nombre d'octets (fenêtre de taille variable).

## Contrôle d'erreur

- Le contrôle d'erreur utilise une stratégie **d'acquiescement positif** avec **temporisateur** associé à une procédure de **retransmission**.

- Son originalité réside dans l'algorithme de calcul de la valeur du temporisateur de retransmission.

TCP doit pouvoir fonctionner avec des réseaux très lents ou très rapides

La valeur du temporisateur armé lors d'une émission dépend du délai d'aller-retour mesuré pendant une période récente.

## Options/Qualité de service

- L'émetteur peut faire appel à la fonction **push** qui force l'émission immédiate des données en attente.

Exemple: pour forcer l'envoi d'une réponse fournie par un opérateur.

- L'émetteur peut définir des données urgentes.

Un bloc urgent est défini à l'intérieur du segment (utilisation du bit URG et du champ pointeur urgent).

Le destinataire est averti dès la réception de l'arrivée de données urgentes.



## TCP: Améliorations

- **Contrôle de congestion** (en fait contrôle d'admission) : algorithme du 'slow start'

- **Adaptation aux temps de réponse** : mesure du délai d'aller retour => adaptation des délais de garde.

- **Gestion de taille des segments**

Problème : 'Silly Window Syndrome'

Emission éventuelle de très petits messages des qu'un émetteur reçoit un crédit.

Différer les émissions tant qu'il y a des données non acquittées

Emettre lorsque tout a été acquitté ou qu'un message de taille suffisante existe.

## **Conclusion**

TCP est un protocole de transport fiable complexe et puissant.

Il a rendu et rendra des services indispensables dans le domaine des réseaux basses et moyennes vitesses.

### **Problèmes posés**

#### **Passage aux réseaux gigabit.**

Différentes expériences et des travaux de recherche sont conduits pour apporter des réponses à cette question.

#### **Support des applications multimédia**

Nécessite la définition de nouveaux protocoles de transports qui respectent les besoins de qualité de service de ces applications (RTP ,.....)

**3**

**Exemple de service TCP/UDP:**

**L'interface SOCKET Berkeley**

## Généralités interface "socket"

Définie en 1982 comme interface de programmation d'applications réseaux (API) pour la version UNIX Berkely (BSD).

Existence de plusieurs autres interfaces d'accès possibles (TLI, NETBEUI, ...)

### Objectifs généraux

. Fournir des moyens de communications entre processus (IPC) **utilisables en toutes circonstances**: échanges locaux ou réseaux.

. Cacher **les détails d'implantation** des couches de transport aux usagers.

. Si possible cacher les **différences entre protocoles de transport hétérogènes** sous une même interface (TCP, Novell XNS, OSI)

. Fournir une interface d'accès qui se rapproche **des accès fichiers pour simplifier la programmation**.

=> En fait des similitudes et des différences majeures entre sockets et fichiers.

## Choix de conception des sockets

Une "socket" (prise) est un point d'accès de service pour des couches transport essentiellement TCP/UDP mais aussi d'autres protocoles (OSI, DECNET...).

- La caractéristique principale d'une socket est donc son **type**:

Pour quel protocole de transport est-elle un point d'accès de service?

Quelle est la sémantique de l'accès de service?

- Une socket possède un nom: un identifiant unique sur chaque site (en fait un entier sur 16 bits) appelé "**numéro de port**".

- Une socket est caractérisée par un **ensemble de primitives** de service pour l'accès aux fonctions de transport.

- Une socket encapsule des données:

**un descriptif** (pour sa désignation et sa gestion)

**des files d'attente** de messages en entrée et en sortie.

## Désignation des sockets

- Pour identifier complètement une socket dans un réseau et pour une couche transport il faut un couple qui l'identifie de façon unique dans ce réseau pour ce transport:

Exemple Internet avec TCP:

( N° Port TCP , @IP )

(Numéro de port TCP , Adresse IP)

- Certains numéros sont réservés pour des services généraux et sont des ports **bien connus** ou "well-known ports".

Exemples: Sockets UDP

Echo server: 7,

DNS: 53,

TFTP: 69.

Exemples: Sockets TCP

FTP: 21,

Telnet: 23,

DNS: 53,

HTTP: 80

## Choix de conception des sockets avec TCP

- TCP est un transport **fiable** en **connexion** et en mode **bidirectionnel point à point**.
- Une socket TCP peut être utilisée par **plusieurs connexions** TCP simultanément.
- Une **connexion est identifiée par le couple** d'adresses socket des deux extrémités.
- Un échange TCP est orienté **flot d'octets**.  
Les zones de données qui correspondent à des envois successifs ne sont pas connues à la réception.

Pour optimiser TCP peut tamponner les données et les émettre ultérieurement.

L'option "**push**" permet de demander l'émission immédiate d'un message.

L'option "**urgent**" permet l'échange de données exceptionnelles avec signalement d'arrivée.

## Choix de conception des sockets avec UDP

- UDP est une couche transport **non fiable**, **sans connexion**, en mode **bidirectionnel** et **point à point**.

- L'adresse UDP d'une socket sur l'Internet est identique à celle d'une socket TCP.

Rappel: les deux ensembles d'adresses sont indépendants:

( N° Port UDP , @IP )

(Numéro de port UDP , Adresse IP)

- Un échange UDP est sans connexion (échange de **datagrammes**).

- Les zones de données qui correspondent à des envois successifs sont **respectées** à la réception.



## Les primitives de l'interface socket

Exemples en langage C sous UNIX.

### socket

- Permet la création d'un nouveau point d'accès de service transport:
  - définition de son type.
  - allocation de l'espace des données.
  
- Trois paramètres d'appel
  - . **"Famille"** d'adresses réseaux utilisées locale, réseau IP, réseau OSI ...
  - . **Type** de la socket (du service) sémantique de la communication.
  - . **Protocole** de transport utilisé.
  
- Un paramètre résultat:
  - le numéro de descripteur socket.

- Profil d'appel de la primitive en C

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
int socket ( int famille,
            int type,
            int protocole);
```

# Aprofondissement des paramètres de la primitive socket

## Paramètre Famille

AF\_UNIX : Communication locale (i-node)  
AF\_INET : Communication Internet  
AF\_ISO : Communication ISO  
....

## Paramètre Type

- **SOCK\_STREAM** : Flot d'octets en mode connecté  
(ne préserve pas les limites de l'enregistrement)
- **SOCK\_DGRAM** : Datagramme en mode non connecté  
(préserve les limites de l'enregistrement)
- **SOCK\_RAW** : Accès aux couches basses.
- **SOCK\_SEQPACKET** : Format structuré ordonné  
(protocoles différents de l'Internet)

## Paramètre Type de protocole

Valeur	Relation avec le paramètre type
IPPROTO_TCP	SOCK_STREAM
IPPROTO_UDP	SOCK_DGRAM
IPPROTO_ICMP	SOCK_RAW
IPPROTO_RAW	SOCK_RAW

## bind

- Primitive pour l'attribution d'une adresse de socket à un descripteur de socket.
  
- Ceci n'est pas fait directement lors de la création du descriptif.
  - . Un serveur (qui accepte des connexions) doit définir sur quelle adresse.
  - . Un client (qui ouvre des connexions) n'est pas forcé de définir une adresse (qui est alors attribuée automatiquement).

### - Profil d'appel de la primitive

```
#include <sys/types.h>
#include <sys/socket.h>

int bind (int s,
          struct sockaddr_in *mon_adresse,
          int longueur_mon_adresse)
```

- Trois paramètres d'appel
  - . Numéro du descriptif de Socket (s).
  - . Structure de donnée adresse de socket  
Pour internet type `sockaddr_in`.
  - . Longueur de la structure d'adresse.

## Aprofondissement concernant la primitive bind

- Descripteur d'adresse de socket pour les protocoles Internet.

```
#include <sys/socket.h>
struct sockaddr_in
{
    short          sin_family;
    u_short        sin_port;
    struct in_addr sin_addr;
    char           sin_zero[8];
};
```

- Un exemple d'exécution de "bind" pour les protocoles Internet.

```
struct servent      *sp
struct sockaddr_in  sin
/* Pour connaître le numéro de port */
if((sp=getservbyname(service,"tcp")==NULL)
    Cas d'erreur
/* Remplissage de la structure sockaddr */
/* htonl convertit dans le bon ordre */
/* INADDR_ANY adresse IP du site local */
sin.sin_family= AF_INET;
sin.sin_port = sp -> s_port;
sin.sin_addr.s_addr=htonl(INADDR_ANY):
/* Création d'une socket internet */
if ((s=socket(AF_INET,SOCK_STREAM,0))<0)
    cas d'erreur
/* Attribution d'une adresse */
if (bind(s, &sin, sizeof(sin)) < 0)
    cas d'erreur
```

## listen

- Utilisé dans le mode connecté lorsque plusieurs clients sont susceptibles d'établir plusieurs connexions avec un serveur.

- Celui ci indique le nombre d'appel maximum attendu pour réserver l'espace nécessaire aux descriptifs des connexions.

- La primitive listen est immédiate (non bloquante).

- Profil d'appel

```
int listen(int s , int max_connexion)
```

`s` : Référence du descripteur de socket  
`max_connexion` : Nombre maximum de connexions.

## accept

- Dans le mode connecté la primitive **accept** permet de se bloquer en attente d'une nouvelle demande de connexion

- Après l'accept, la connexion est complète entre les deux processus.

- Le site qui émet accept exécute une ouverture passive.

- Pour chaque nouvelle connexion entrante la primitive fournit un pointeur sur une nouvelle socket qui est du même modèle que la socket précédemment créée.

### - Profil d'appel

```
#include <sys/types.h>
#include <sys/socket.h>
int accept
    (int ns,
     struct sockaddr_in *addr_cl,
     int lg_addr_cl)
ns          : Référence nouvelle socket
addr_cl    : L'adresse du client.
lg_addr_cl: La longueur de l'adresse.
```

## Aprofondissement concernant les primitives listen et accept

Exemple de code pour un serveur qui accepte des connexions successives et qui crée un processus pour traiter chaque client.

```
#include <sys/socket.h>

/* Adresse socket du client appelant */
struct sockaddr_in from;

quelen = ... ;
if (listen (s, quelen) <0 )
    Cas d'erreur

/* On accepte des appels successifs */
/* Pour un appel on crée un processus */

if ((g=accept(f, &from, sizeof(from))) <0)
    Cas d'erreur

if ( fork ...
/* Processus traitant de connexion*/
```

## connect

- La primitive **connect** (bloquante) permet à un client de demander l'ouverture (**active**) de connexion à un serveur.

- L'adresse du serveur doit être fournie. La partie extrémité locale relative au client est renseignée automatiquement.

- Pour un échange connecté, le connect permet d'utiliser ensuite les primitives read, write, send, recv.

Le client ne doit plus fournir l'adresse du serveur pour chaque appel mais le descriptif de la socket..

### - Profil d'appel

```
#include <sys/types.h>
#include <sys/socket.h>
int connect
    (int s,
     struct sockaddr_in *addr_serv,
     int lg_addr_serv)
```

s : La référence de la socket

addr\_serv : L'adresse du serveur.

lg\_addr\_serv : La longueur de l'adresse.



## send, recv

- Les primitives **send**, **recv** (bloquantes) permettent l'échange effectif des données.

- Le profil d'appel est identique à celui des primitives `read` et `write` sur fichiers avec un quatrième paramètre pour préciser des options de communications.

### - Profil d'appel

```
#include <sys/types.h>
#include <sys/socket.h>
```

```
int send
    (int s, char *zone,
     int lg_zone,int options)
```

```
int recv
    (int s,char *zone,
     int lg_zone,int options_com)
```

`s` : La référence de la socket  
`zone` : La zone à échanger.  
`lg_zone` : La longueur de la zone.  
`options_com` : Les options (données urgentes , ....)

## sendto, recvfrom

- Les primitives **sendto**, **recvfrom** permettent l'échange des données plutôt dans le mode non connecté UDP.

- On doit préciser l'adresse destinataire dans toutes les primitives **sendto** et l'adresse émetteur dans les **recvfrom**.

### - Profil d'appel

```
#include <sys/types.h>
#include <sys/socket.h>
int sendto (
    int s,
    char *zone,
    int lg_zone,
    int options_com,
    struct sockaddr_in *addr_dest,
    int lg_addr)
int recvfrom (
    int s,
    char *zone,
    int lg_zone,
    int options_com,
    struct sockaddr_in *addr_emet,
    int *lg_addr)
```

**addr\_dest** : L'adresse du destinataire.

**addr\_emet** : L'adresse de l'émetteur.

**lg\_addr** : La longueur de l'adresse.

## Shutdown

- Permet la purge des données en instance sur une socket avant la fermeture.

```
shutdown(s , h);
```

h = 0 l'utilisateur ne veut plus recevoir de données  
h = 1 l'utilisateur ne veut plus envoyer de données  
h = 2 l'utilisateur ne veut plus ni recevoir, ni envoyer.

## close

- Permet la fermeture d'une connexion et la destruction du descriptif.

- Profil d'appel

```
#include <sys/types.h>
#include <sys/socket.h>

int close ( int s )
```

## **Fonctionnement en TCP**

- Serveur.

socket  
bind  
listen  
accept  
recv, send  
close

- Client.

socket  
connect  
recv, send  
close

## **Fonctionnement en UDP**

socket  
recvfrom, sendto  
close