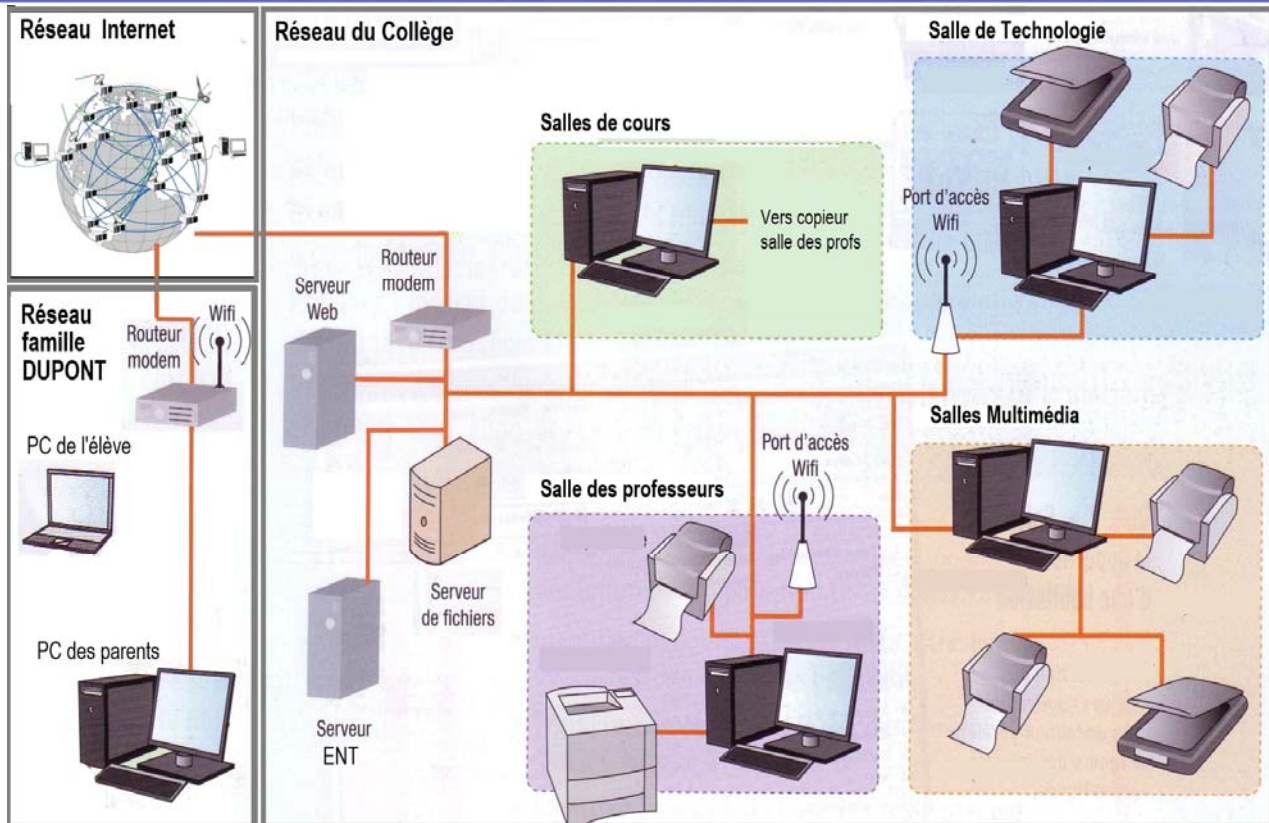


# Introduction



# Introduction





## ■ Besoins / applications

⇒ Aujourd'hui : besoin de pouvoir échanger tout type d'information : données (texte, nombres, schémas...), son, image, vidéo (images animées)

⇒ Besoins à satisfaire et contraintes à respecter différentes selon les informations à transmettre et le domaine d'application :

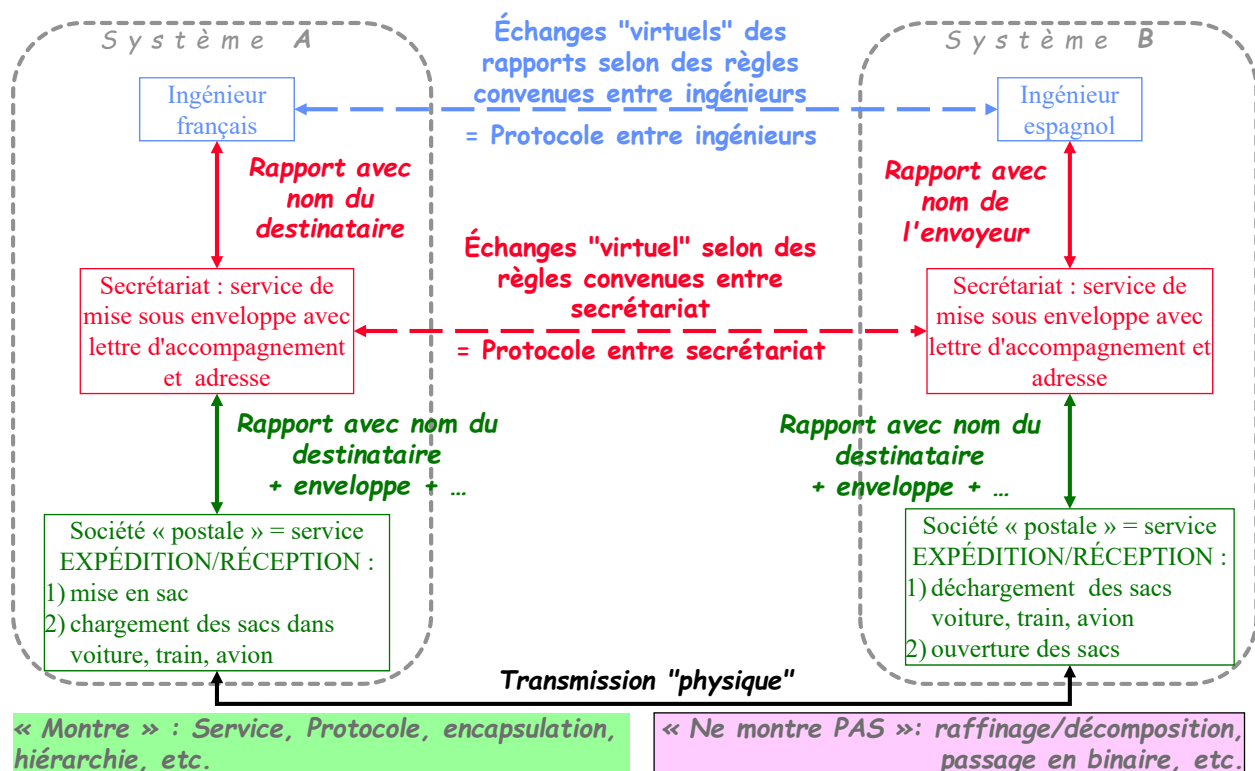
- **Téléphonie** (voie seulement) : orienté de connexion, débit faible (64 kbits/s : un échantillon sur 8 bits 8000 fois par s), sensible au temps de réponse mais supporte des pertes/altérations non négligeables.
- **Données informatiques** (ex. : fichiers) : orienté connexion ou non, débits variables, moins sensible au temps de réponse mais ne supporte pas de pertes/altérations.
- **Vidéoconférence** : orienté connexion, débits très élevés, très sensible au temps de réponse
- **Systèmes de commandes** : sensibilité très critique au temps de réponse → temps réel dur et non plus de confort, et toute perte/altération peut être catastrophique.

DÉBIT, BANDE PASSANTE, TEMPS DE RÉPONSE,  
QUALITÉ DE SERVICE, ÉTABLISSEMENT DE CONNEXION ou non, SÉCURITÉ,

# Introduction



## ■ Exemple de quelques principes d'échanges d'information



# Introduction

## ■ Notions identifiées

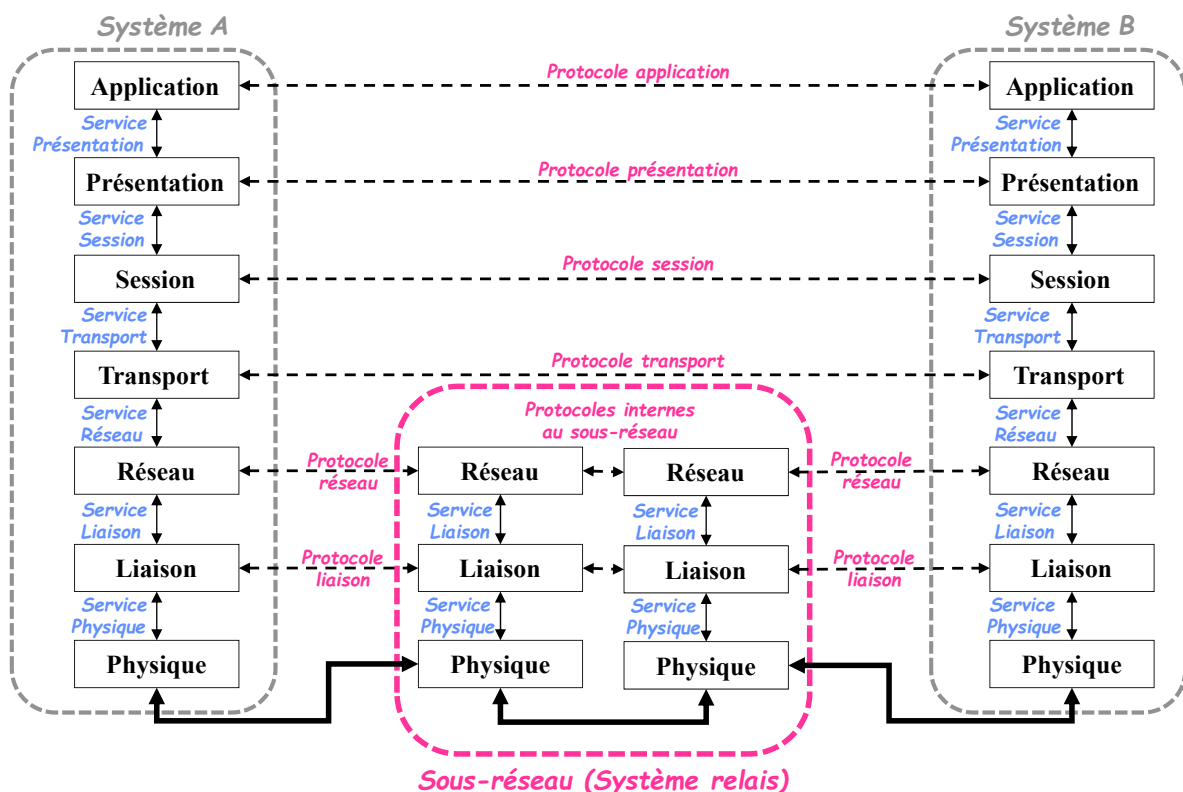
- fonctionnalités avec **différents niveaux D'ABSTRACTION** → existence d'une **hiérarchie**
- des « **SERVICES** » pour les « échanges » entre 2 niveaux différents : un niveau utilise les services fournis par un niveau inférieur, et fournit des services à un niveau supérieur
- des **PROTOCOLES** (règles, formats) pour les « échanges » entre niveaux de même abstraction
- l'information est **ENCAPSULÉE / DÉSENCAPSULÉE** entre 2 niveaux d'abstraction différents

⇒ Hiérarchiser pour dégager les fonctions principales nécessaires à la gestion d'un réseau

⇒ Normaliser aussi

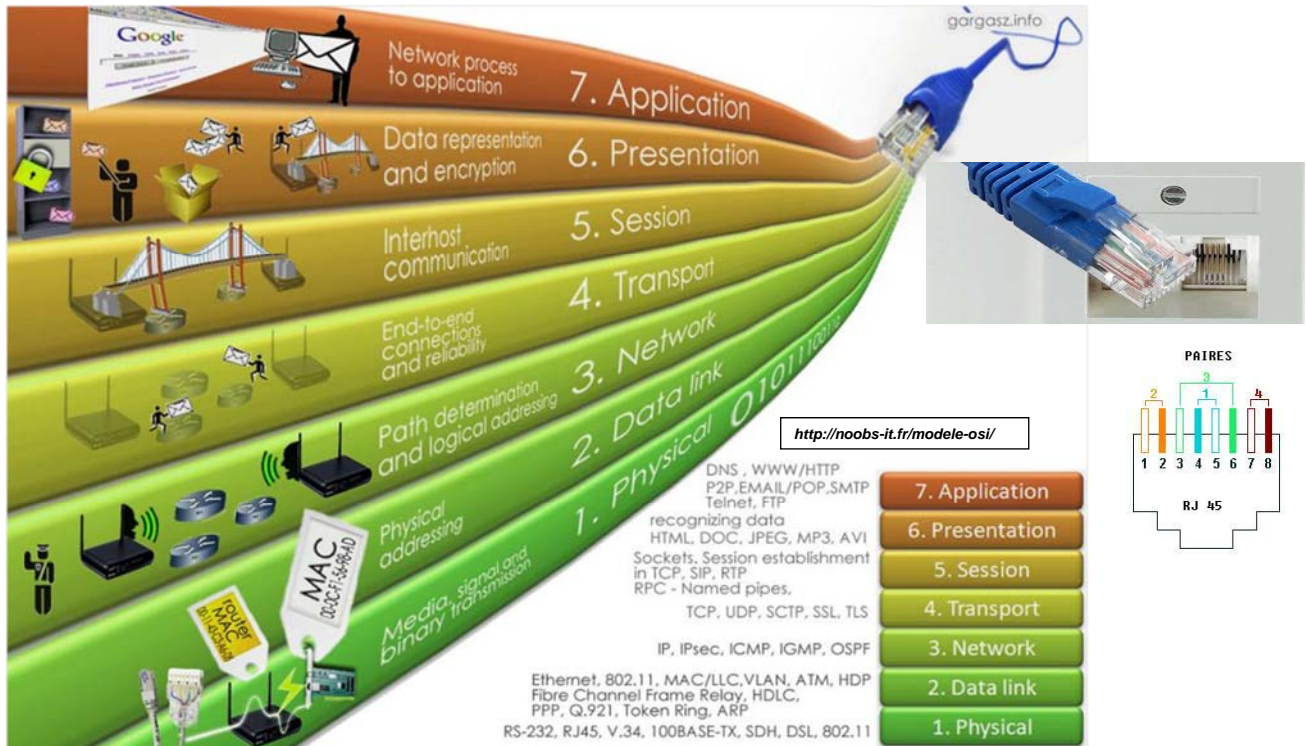
# Introduction

## ■ Ce que cela donne dans le modèle OSI global



# Introduction

⇒ Joli ... mais des effets visuels trompeurs



## Éléments Bibliographique

### ■ Livres :

- **Réseaux Informatiques : Cours et Exercices, tome 1 (de la transmission de données à l'accès au réseau), tome 2 (du transport des données aux applications).** Danièle DROMARD, Fetah OUZZANI, Dominique SERET (Ed. Eyrolles)
- **Réseaux locaux informatiques :** G. PUJOLLE, M. SCHWARTZ (Editions Eyrolles)
- **Initiation aux Réseaux : "cours" et exercices.** Guy Pujolle (Ed. Eyrolles), octobre 2000, ISBN : 2 212 09155 9
- **Réseaux et télécoms : "cours" et exercices corrigés.** Claude SERVIN (Ed. DUNOD, série Sciences SUP), juin 2003, ISBN : 2 10 007986 7
- **Les réseaux : principes fondamentaux :** Pierre ROLIN, Gilbert MARTINEAU, Laurent TOUTAIN, Alain LEROY (Ed. Hermes 1997)
- **Réseaux : Architecture, protocoles, applications.** A. TANENBAUM (Inter EDITIONS)
- **Clients/Serveurs : guide de survie.** Robert ORFALI, Dan HARKEY et Jery EDWARDS. 3ème édition (Ed. Vuibert 1999)

### □ Sites :

- <http://kadionik.vvv.enseirb-matmeca.fr/>
- <http://www.machaon.fr/reseaux.html>
- <http://www.irisa.fr/prive/bcousin/reseaux-generalites.html>
- <http://www.guill.net/index.php?cat=12>
- ..... et beaucoup plus encore.

### □ Outils :

- Une série d'outils <http://network-tools.com/>
- Suivi de "chemin IP" : <http://www.monitis.com/traceroute/>
- Un analyseur de trafic : <https://www.wireshark.org/>
- **Récupérer des fichiers – RECUVA :** <http://www.commentcamarche.net/download/telecharger-34055403-recuva>
- **Nettoyer sa machine - CCLEANER :** <http://www.commentcamarche.net/download/telecharger-168-ccleaner>



---

## - 1<sup>ère</sup> partie -

### Réseaux de communications : historique et concepts de base

---

## 1 - Historique et contexte

### ■ 1. Historique des réseaux de communication (de type informatique) : d'un ordinateur aux systèmes distribués et aux réseaux

⇒ **Années 1940 : 1er calculateur électronique** → hors de prix cher, ultra volumineux, fragile, et surtout, isolé et rudimentaire : ni écran, ni clavier, ni mémoire, ....

⇒ Évolutions guidées par 2 axes de progrès souvent fortement corrélés :

- **axe technologique :**
  - **MATÉRIEL** : électronique, processeurs, mémoires, écrans, X-G, fibre, etc.
  - **LOGICIEL** (langages et concepts de programmation)
- **axe applicatif** : applications possibles et/ou « rêvées », à faire exécuter par les calculateurs

#### 1.1. Émergences des premiers besoins

Ajout de périphériques "locaux" au calculateur : **terminal (écran + clavier), mémoire, ...**

⇒ *ordinateur = ensemble d'entités "communicantes"*

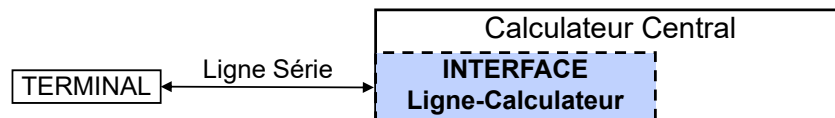
⇒ Mise en lumière des problèmes de base pour communiquer :

- ① **Nature et Structuration des Supports de communication** → **TOPOLOGIE**
- ② **Règles pour l'échange des informations** → **PROTOCOLE DE COMMUNICATION**

## 1.2. "Accès à distance" (Années 60/70)

### a) Pouvoir accéder à distance à UN ordinateur et à SES ressources

- ⇒ **Intérêts** : éviter d'avoir à être physiquement à côté du ordinateur
- ⇒ **Début de solution → UN terminal** "déporté" (à distance) et relié au ordinateur par une ligne de communication (en général série)
  - "télétraitement" : aucun calcul déporté sur le terminal, mais seulement les demandes et les réponses ("résultats")
  - **Pb direct** : la distance et donc le temps de transmission ⇒ **traitements et matériels supplémentaires dédiés à la gestion des communications, mais encore placés dans le ordinateur central**
  - **Schéma : Solution de base à UN Terminal**

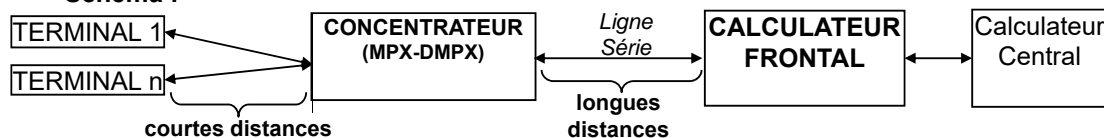


- **Évolution** : possible pour 1 terminal !!!! alors pourquoi pas plusieurs ???

### ⇒ **Evolution → Réseau de terminaux** : plusieurs terminaux pour UN même ordinateur

- **Pb direct** : coûts en gestion des communication ↗ très fort fortement ⇒ le ordinateur central « ne fait plus que ça » ⇒ pour lui rendre son rôle, mise en place :
  - d'un ordinateur supplémentaire, uniquement pour gérer les communications
  - de multiplexeurs pour "rentabiliser" (coût et temps) les lignes

#### – Schéma :



- **Multiplexage** : une solution pour partager une « ligne » ⇒ problématique du **PARTAGE** des RESSOURCES DE COMMUNICATION (medium/ligne ici)

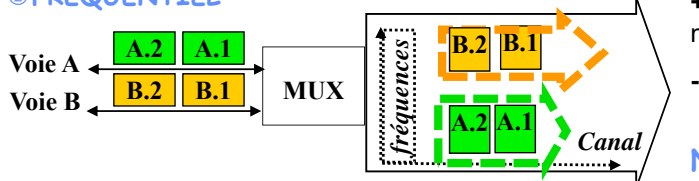
#### ① TEMPOREL



+ : moins coûteux

- : transmission série donc temps de transfert plus long

#### ② FRÉQUENTIEL



+ : transmission parallèle de plusieurs messages ⇒ Full Duplex possible

- : plus coûteux car modulation et filtrage et besoin de plus de bande passante

**Notion de canal Physique / canal Logique**

### 1.3. "Davantage de puissance à moindre coût" (Années 80)



Performances des ordinateurs ↗ et coût ↘ : ils rentrent dans les mondes industriel et privé

#### a) Disposer de plus de puissance à moindre coût

⇒ Plus de puissance = vers la "**DISTRIBUTION/répartition**" du traitement de l'information.

- Ex. : répartir/distribuer un programme sur plusieurs calculateurs monoprocesseurs

⇒ Moindre coût = vers le "**PARTAGE**" des ressources "**INFORMATIQUES**"

- Ex. : partager l'utilisation d'une imprimante entre plusieurs calculateurs

⇒ Appliquer ces 2 principes à plusieurs calculateurs ➡ **réseaux d'ordinateurs ... et ... la communication devient cruciale et une problématique à part entière.**

⇒ Réseaux **HÉTÉROGÈNES** en termes de matériel, systèmes d'exploitation, applications, organisation ... Ex. phare de réseau local : ETHERNET (développé par DEC, Intel et Xerox)

#### b) Nouvelles générations d'applications

⇒ **Messagerie** = échanger quelques **Koctets à l'origine**, mais aujourd'hui volumes au moins  $10^3$  beaucoup plus importants (son, image → des Mo, Go). Service à faible contrainte de temps de réponse (différé) et n'exigeant pas nécessairement de gros débits.

⇒ **News** : extension du mail mais volumes beaucoup plus importants et contraintes de temps plus fortes (pour mise à jour et maintien de cohérence entre les sites dupliqués).

⇒ **Transfert de fichiers** (*ftp* dans internet, FTAM dans l'ISO) : gros volumes en très peu de temps car interactivité entre la demande et la réponse

⇒ **Calculs et commandes répartis ....**

### 1.4. Clients/serveurs et multimedia (Années 90 ...)

**Réseaux désormais partout, utilisés/accessibles par beaucoup de monde et de domaines : Internet, réseaux dans nos voitures, satellites, avions ...**

a) Nouveaux besoins applicatifs : échanger tout type d'information : donnée (texte, ...), son, image, vidéo (images animées), mais besoins/contraintes différents selon les informations et le domaine d'application : **DÉBIT, BANDE PASSANTE, TEMPS DE RÉPONSE, QUALITÉ DE SERVICE, ÉTABLISSEMENT DE CONNEXION ou non, SÉCURITÉ, ...**

#### b) Concept Clients/Serveurs → quelques mots ...

⇒ Modèle Client/Serveur : environnement **LOGICIEL** ouvert et souple permettant de

- faire interagir des systèmes et des applications totalement hétérogènes,
- répartir l'intelligence et la charge de travail, et de dissocier les fonctionnalités

⇒ Beaucoup de visions et de définitions → quelques caractéristiques communes

- **Services par messages** : relation entre des processus, actifs sur des machines différentes (ou pas). Le processus serveur fournit des services à la demande de processus clients, via des **messages**
- **Contrôle d'accès aux ressources** : un serveur contrôle l'accès des clients à des ressources

• **Autres caractéristiques :**

- **protocoles asymétriques** : c'est toujours le client qui déclenche le dialogue, **ET le flux de demandes est beaucoup moins important que le flux de réponse,**
- **Multiplateformes** : un logiciel C/S est indépendant des plateformes

⇒ Ex. : serveurs de fichiers, de comptes, d'applications **WEB**, d'impression, etc.

## ■ 2. Systèmes distribués - Systèmes de communication

Le principe de traitement de l'information (plus généralement d'intelligence) distribué/réparti conduit à 2 notions, à ne pas confondre, bien que dépendantes.

### 2.1. Systèmes et applications distribuées

⇒ **DISTRIBUÉ** versus **RÉPARTI** : faux débat ????

⇒ Définitions :

- répartition de différentes "fonctions" constituant un système ou une application, sur plusieurs "éléments" matériels/logiciels.
- autre définition : ensemble de systèmes numériques intelligents, géographiquement distribués, et qui coopèrent pour réaliser une ou plusieurs tâches.

⇒ Exemples

- calcul numérique sur une machine multiprocesseurs.
- système d'exploitation UNIX : "serveurs" de mails, licences, comptes, ...
- commande d'une chaîne d'assemblage
- tout les systèmes basés "client/serveur"

⇒ Objectifs :

- accroître la rapidité de traitement en partageant des ressources distantes
- limiter l'impact de pannes et assurer un service dégradé (ex : plus de mail, mais comptes encore utilisables)

### 2.2. Système (réseau) de communication

⇒ **Définition** : ensemble de ressources matérielles et logicielles dédiés au transfert de l'information entre au moins 2 entités "intelligentes" distantes pouvant être "automates".

⇒ **Complexité** du système → d'autant plus grande que :

- ① l'information initiale à transférer est complexe ET a une forme éloignée de celle admissible par le support de communication (ex. : transformation de texte et de schémas en signaux sinusoïdaux)
- ② la distance émetteur/récepteur est grande → liaison directe impossible → besoin d'intermédiaires
- ③ l'émetteur et le récepteur sont hétérogènes (en matériel, logiciels, ...)

⇒ **Exemples** : ligne série, ETHERNET, etc. + adaptateurs lignes + cartes réseaux + drivers, ...

⇒ **Objectifs** :

- rendre transparent aux applications (logicielles et humaines), tous les mécanismes nécessaires à leur communications
- être indépendant vis-à-vis des applications qui les utilisent (à relativiser ...)
- réduire les coûts des communications, et les fiabiliser (qualité de service : QoS)
- partager les ressources pour les applications

### 2.3. Systèmes distribués et Réseaux de communication

⇒ Un système distribué utilise nécessairement des réseaux de communications

⇒ Un réseau est par définition lui-même un système distribué

⇒ Mais les problématiques sont en grande partie différentes



### ■ 1. Diversité des réseaux - Normes et standards

#### 1.1. Classification des réseaux

##### a) Critères de classification

⇒ Organisation et protocoles des réseaux définis pour satisfaire des besoins de :

- coût : matériel, logiciel
- temps de réponse
- modularité : maintenance, facilités d'extension et de "fiabilisation"
- qualité de service (QoS) = capacité à délivrer ses services en cas de
  - Surcharge en sollicitations
  - « Défaillances » et « d'attaques »

⇒ Pb : ces critères sont

- antagonistes pour un même contexte
- variables d'un contexte à l'autre

⇒ Aspect commercial ⇒ chaque « constructeur » développe « ses » solutions

⇒ Bilan : trop grande diversité de solutions ⇒ tentatives d'homogénéisation ...

→ NORMALISATIONS ...

##### b) Une classification usuelle : par rapport à la distance

###### ① LAN : Local Area Network

- de quelques dizaines de m à des km : d'une pièce à un ou plusieurs bâtiments
- facilité de hauts débits (jusqu'aux Gbit/s), et **bornage des délais de transmission** ...
- réseaux locaux, dont réseaux locaux Industriels, réseaux embarqués, ...
- ex. : réseau CAN (inférieur à 40m), réseau local de l'Université

###### ② MAN : Metropolitan Area Network

- de quelques km à des centaines de km : échelle d'une ville ou d'un pays (USA ...)
- débits élevés moins faciles
- sert aussi à l'interconnexion de réseaux locaux
- ex. : AQUAREL, REMIP (Internet sur région Aquitaine / Midi-Pyrénées), RENATER (Internet en France pour le monde de la recherche)

###### ③ WAN : Wide Area Network

- d'un pays à la planète
- débits élevés encore moins faciles

##### c) Autres classifications

⇒ Réseaux sans/avec fils, réseaux Hauts/Faibles débits, ....

## 1.2. Normes et standards (le mot "anglais" pour « norme » est « Standard »)



### a) Normes

- ⇒ Document technique précisant les caractéristiques d'un type de matériel ou de logiciel déterminé, visant à homogénéiser les règles pour faciliter l'interconnexion des éléments des réseaux.
- ⇒ règles "ouvertes" (pas sous dictées par un constructeur) pour être exploitables par tous les constructeurs et utilisateurs pour proposer des solutions
- ⇒ PB : plusieurs "communautés" ont des intérêts à la normalisation, mais pour des raisons différentes ...
  - **Gouvernements** : pour garantir la "libre concurrence", mais aussi, à l'opposé, pour favoriser l'industrie de leur pays, et leur prédominance technologique (marchés publics)
  - **Constructeurs** : gagner des parts de marché
  - **Utilisateurs** : faire respecter leurs besoins et leur indépendance

⇒ multitude d'organismes de normalisation : gouvernementaux/privés, internationaux/nationaux

### b) Principaux organismes de normalisation

- ⇒ **ISO : International Standard Organization**
  - non gouvernemental, mais officiel et reconnue par l'ONU, regroupant une centaine de pays
  - couvre tous les domaines techniques sauf : électricité, électronique et électrotechnique (couverts par la CEI : Commission Electrotechnique Internationale)
  - harmonisation ISO-CEI -> JTC1 (Joint Technical Comitee 1)
- ⇒ **IEEE : Institute of Electrical and Electronics Engineers**
  - organisme américain non officiel, mais à l'origine des principales normes des réseaux locaux

- ⇒ **ITU : International Telecommunication Union**
  - inter-gouvernementale reconnue par l'ONU
  - créée en 93 : regroupe CCIR + IFRB + CCITT
- ⇒ **ANSI : American National Standard Institute**
- ⇒ **AFNOR : Association Française de NORmalisation**
- ⇒ Organismes Européen : CEN, CENELEC, ETSI, ...

### c) Standards

- ⇒ Matériel ou logiciel dont la diffusion est tellement importante qu'il devient une référence dans son domaine (parfois incontournable), même s'il ne répond pas aux normes en vigueur (ex. : Internet)
- ⇒ Un standard est issu d'un constructeur, ou regroupement de constructeurs

## ■ 2. Problèmes et terminologie de base

### 2.1. Terminologie : Réseau = machines-hôtes + sous-réseau

- ⇒ Principalement issue d'ARPANET (ancêtre d'Internet), projet de l'armée américaine
  - **Machines-hôtes** = "ordinateurs" connectés au réseau avec donc, en plus, les logiciels/matériels de mise en œuvre des communication
  - **Sous-réseau** = lignes/canaux/voies de communication + éléments de commutation
    - élément/nœud de commutation = assure/gère uniquement des fonctions « réseaux »
    - canal = lien entre des éléments de commutation ou machines-hôtes
- ⇒ Aujourd'hui
  - nœud = une machine-hôte ou un élément de commutation
  - canal = lien entre 2 nœuds
  - abonnés = tout équipement connecté à un réseau, avec des fonctionnalités autres qu'uniquement réseau
  - réseau = ensemble de nœuds et abonnés reliés par des canaux

### 2.2. Notions de base à définir dans les réseaux

- ⇒ Nature de ligne/support physique ... canal/voie logique
- ⇒ Adaptation de la forme de l'information au type de support
- ⇒ Topologie = organisation physique et logique des éléments du réseau
- ⇒ Adressage des abonnés/nœuds du réseau pour pouvoir les « reconnaître »
- ⇒ Commutation/Routage = découpage de l'information / choix du chemin pour la transférer
- ⇒ Connexion ou non entre les abonnés
- ⇒ Contrôle/correction des flux et des contenus des messages

## ■ 3. Supports et Canaux de communication



### 3.1. Canal physique

#### a) Définitions

- ⇒ Support physique qui véhicule **réellement** l'information entre 2 nœuds d'un réseau
- ⇒ Selon le nombre de nœuds reliés à un canal, on distingue :
  - canal point à point : relie uniquement 2 nœuds ⇒ lien personnel (privé) entre 2 nœuds



- canal multipoints : relie au moins 3 nœuds ⇒ notion de **DIFFUSION**



- ⇒ Composition d'un canal :
  - canal simple = lien direct entre abonnés sans passer par des nœuds intermédiaires
  - canal composé = lien indirect entre abonnés → passage par des nœuds intermédiaires

#### b) Les supports physiques (cf. chapitre correspondant)

- ⇒ Différents types aux caractéristiques différentes : débit, bande passante, coût, atténuation, sensibilité aux perturbations, etc. : **paire bifilaire torsadée (aluminium ou cuivre), câble coaxial, fibre optique, ondes en air libre : infra rouge, hertziennes, ...**

### 3.2. Canal logique

- ⇒ Tout découpage LOGIQUE d'un canal PHYSIQUE. Ex. : faire du multiplexage temporel sur un canal, c'est créer "artificiellement" l'équivalent de plusieurs canaux

## ■ 4. Les différentes topologies

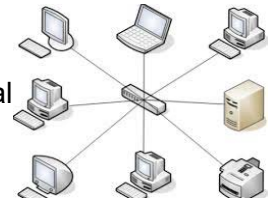


Topologie = organisation/structuration/mise en relation des différents éléments du réseau

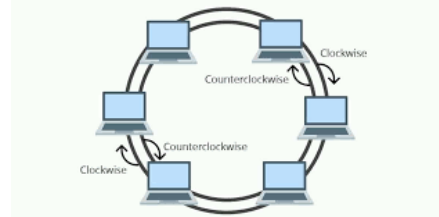
### 4.1. Topologie physique = organisation physique

#### a) Basée sur des canaux points à points

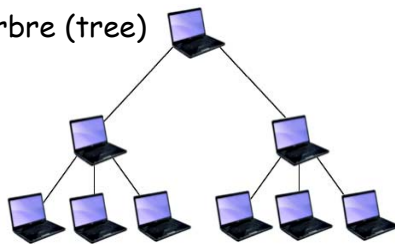
⇒ Étoile (star) : toutes les communications passent par un nœud central (sans que celui-ci supervise nécessairement celui la gestion)



⇒ Anneau (ring) simple (double) : chaque nœud a un suivant (et un précédent) → notion d'ordre



⇒ Arbre (tree)

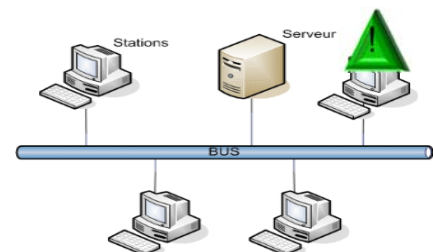


Maillé (mesh) ..



#### b) Basée sur des canaux multipoints : diffusion

⇒ BUS : toutes les communications passent par le même canal sur lequel sont connecté les nœuds



⇒ Radio/Satellites :



### 4.2. Topologie logique = organisation logique

⇒ C'est l'organisation « LOGIQUE/VIRTUELLE » des différents éléments du réseau.

⇒ Ex. : un bus physique peut-être gérée comme s'il s'agissait d'une étoile, et réciproquement

### 4.3. Choix d'une Topologie

⇒ Répond à de multiples critères, dont la distance (pour des raisons de temps de réponse), la fiabilité, modularité, sécurité, ...

## ■ 5. Adressage des éléments du réseau



### a) Besoins d'adressage

⇒ **Adressage du DESTINATAIRE** : une information peut-être à envoyer à :

- ① un seul destinataire
- ② un groupe de destinataires
- ③ tout le monde

Dans les cas ① et ②, il faut connaître l'@ du ou des destinataires ⇒ chaque élément "actif" du réseau doit se voir attribué une adresse unique pour pouvoir être identifié

⇒ **Adressage de L'ÉMETTEUR** : un destinataire doit parfois aussi pouvoir répondre et/ou « identifier » l'émetteur et « valider » l'information reçue

⇒ Cas d'adressage inutile : ligne entre 2 calculateurs, diffusion générale (ex. télévision)

⇒ Il existe de multiples systèmes d'adressage, dépendant notamment de la topologie, du constructeur, du niveau d'abstraction de l'information, ...

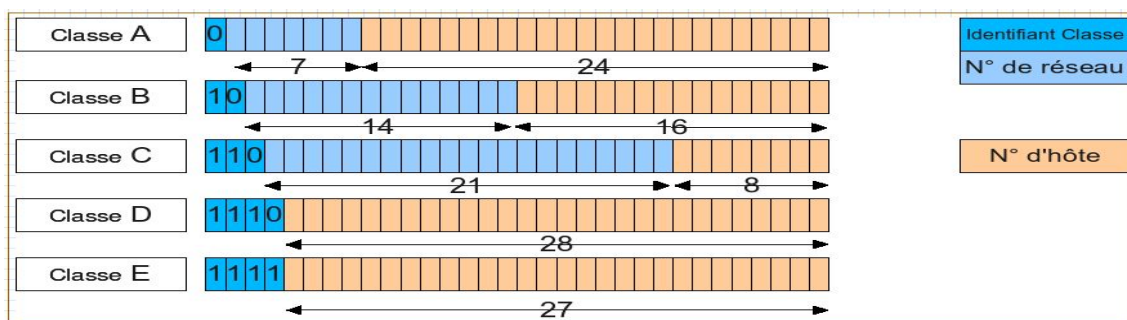
### b) Exemples

⇒ Adresse MAC :

- adresse unique, attribuée à une carte réseau en général
- longueur fixe de 6 octets,
- 3 octets de poids fort identifient le constructeur, et les 3 autres octets donnent le numéro de la carte
- Format typique : ex. → **02:0f:b5:cd:1b:7f**

⇒ Internet : adresse au "niveau IP", dite adresse IP

- une adresse unique, attribuée à une "machine" (Pc, station, routeur, ..)
- longueur fixe de 32 bits, présenté sous forme **XXX.XXX.XXX.XXX** avec XXX un nombre décimal dans l'intervalle [0, 255]. Ex. : 192.168.0.1
- les bits de poids forts déterminent l'adresse du réseau, et les bits de poids faibles l'adresse de la machine dans ce réseau. Ce nombre respectif de bits dépend de la classe
- 4 classes d'adresses (s'adapter aux différentes tailles de réseaux) : A, B, C et étendues



Classes	réseau	hôte	Nombres d'adresses
A 126 réseaux	1 à 126	0.0.1 à 255.255.254	$2^{24}-2=16777214$
B $2^{14}=16384$	128.1 à 191.254	0.1 à 255.254	$2^{16}-2=65534$
C $2^{21}=2100000$	192.0.1 à 223.255.254	1 à 254	$2^8-2=254$



## ■ 6. Contrôle du transfert de l'information



Recevoir au mieux ce qui a été émis ⇒ détecter les "écarts" et les corriger

### 6.1. Transfert d'information : Mode connecté ou non ?

#### a) Mode connecté

⇒ Principe = trouver un chemin (physique ou logique) entre l'émetteur et le récepteur avant de commencer à transférer les informations (ex : téléphonie) --> 3 étapes :

- ① établissement d'un chemin avant d'échanger de l'information et réservation au besoin des ressources de communication nécessaires
- ② échanges sans possibilité d'interruption
- ③ fermeture de la connexion, et relâchement, si besoin est, des ressources

+ : sécurisation du transfert : assurance de l'existence du récepteur

— : lourd et sous-utilisation des ressources réservées durant toute la communication, quelle que soit la "densité" du trafic

— : accès multipoint difficile, car impose d'établir autant de connexion qu'il y a de destinataires

— : temps d'attente induit par la recherche de l'établissement d'une connexion

⇒ ex. : protocole HDLC (un protocole du niveau 2 du modèle OSI)

#### b) Mode NON connecté



⇒ Principe = informations émises sans qu'un chemin entre l'émetteur et le récepteur ait été déterminé (ex : mail).

+ : pas de perte de temps pour l'établissement de la connexion

— : l'émetteur n'a aucune garantie que le récepteur recevra l'information

### 6.2. Contrôle et correction de l'information

#### a) Contrôle d'erreurs

⇒ Mécanismes pour vérifier certaines caractéristiques relatives au contenu des "unités" d'information, et détecter ainsi les unités erronées.

⇒ Ex. : comptage du nombre de bits, valeurs des bits

#### b) Contrôle de flux

⇒ Mécanismes pour vérifier l'ordre et le nombre d'unités, et détecter le déséquencelement ou la perte d'unités + éviter la perte d'unités en "freinant" le flux

#### c) Correction d'erreurs

⇒ Mécanismes pour corriger une partie des erreurs détectées sur le contenu des "unités" d'information (pas sur les problèmes de flux)

**Attention : tout n'est pas détectable, et encore moins rectifiable**

## ■ 7. La commutation d'information

5 catégories (principes) de commutation (depuis la plus ancienne) : **circuits**, **paquets**, **trames** (frame relay), **cellules** (ATM)

### 7.1. Commutation de circuits

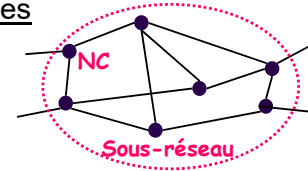
⇒ Développé dans le domaine de la téléphonie (transfert de la voie initialement) où un commutateur est appelé autocommutateur

⇒ Principe : celui du mode connecté appliqué aux circuits physiques

① établissement et réservation d'un chemin physique

② échanges sans possibilité d'interruption

③ relâchement des ressources de communication



⊕ : une fois la communication établie, délai de transfert  $\cong$  temps de propagation du signal : intéressant pour les applications temps réel

⊕ : diminution du risque de pertes de données (mais pas des erreurs sur les données)

⊖ : tous ceux du mode connecté, dont notamment la sous-utilisation des ressources réservées durant toute la communication, dans le cas de trafic sporadique comme l'est la téléphonie.

**Amélioration : autoriser plusieurs communications à partager (simultanément) le même support physique**

- pas de réservation d'autant de ressources de communication
- stocker des données dans les commutateurs obligatoire maintenant

### 7.2. Commutation de messages

⇒ Principe

① pas de réservation d'un chemin physique avant d'échanger de l'information

② les messages sont transmis en TOTALITÉ, de nœud en nœud, jusqu'au destinataire, et chaque nœud doit attendre d'avoir reçu un message intégralement et correctement avant de le retransmettre ⇒ chaque nœud nécessite :

- des buffers
- des mécanismes de contrôle et de reprise des messages :
  - **contenus des messages** : ne pas retransmettre un message faux, et occuper alors inutilement le réseau
  - **flux** : ne pas saturer les buffers et donc ne pas perdre des messages

⊕ : meilleure utilisation du support (optimisation de la bande passante et du temps)

⊕ : pas de ressources physiques à réserver, donc pas de perte de temps

⊖ : coût des commutateurs : buffers de forte capacité et mécanismes de contrôles

⊖ : transferts ralentis par stockage et contrôle à chaque étape

**Amélioration : décomposer un message en plusieurs morceaux : "paquets" ou "trames"**

### 7.3. Commutation de paquets

- ⇒ Paquet = suite d'informations, du niveau 3 (modèle OSI) et de **longueur MAX préfixée**
- ⇒ Principe : celui de la commutation de message, appliqué aux paquets, MAIS :  
les divers paquets d'un même message, ne suivent pas nécessairement le même chemin ⇒ 2 modes
  - ① Connecté (circuit virtuel) : tous les paquets suivent le même chemin prédéterminé :
    - ⇒ pas de déséquencelement dans l'ordre des paquets
    - ⇒ paquets dépendants/liés ⇒ un seul contient @destinataire
    - ⇒ ex. : X25
  - ② NON Connecté (Datagramme) : les paquets suivent des chemins différents :
    - ⇒ déséquencelement des paquets ⇒ ré-ordonnancement nécessaires
    - ⇒ paquets INdépendants ⇒ tous doivent contenir @destinataire + n° du paquet
    - ⇒ ex : TCP/IP sous internet
- + : meilleure utilisation du support (optimisation de la bande passante et du temps)
- + : taille réduite et bornée des paquets ⇒ taille des buffers réduite + gain de temps + moins de risque de perte de paquets
- + : gain de temps, puisque transmission // possible des paquets d'un même message
- : augmentation du trafic
- : mécanismes de contrôles
- : transferts ralentis par stockage et contrôle à chaque étape

### 7.4. Commutation de trames

- ⇒ Trame = suite d'informations, du niveau 2 (modèle OSI) et de longueur max préfixée
- ⇒ Principe : celui de la commutation de paquets, appliqué aux trames, avec 2 modes :
  - ① Commutation de trame (**frame switching**) : mécanismes de contrôle et de reprise d'erreurs mis en œuvre
  - ② relayage de trame (**frame relay**) : pas de mécanismes de contrôle et de reprise d'erreurs mis en œuvre
- ⇒ Remarque : commutation et relayage de trame intègrent les notions d'adressage, de routage et de contrôle de flux, habituellement traitées par le niveau 3 du modèle OSI.

### 7.5. Commutation de cellules

- ⇒ Cellule = paquet de taille fixée à 53 octets (5  $\phi$  de contrôle + 48  $\phi$  de données)
- ⇒ Principe : celui de la commutation de paquets en mode connecté et sans reprise d'erreurs. Très adapté au transfert de l'image, car hauts débits
- ⇒ Ex : ATM

## ■8. ROUTAGE de l'information

- ⇒ Mécanisme qui permet de choisir le meilleur chemin pour transférer une information dans le réseau
- ⇒ Élément actif spécialisé dans cette tâche = un routeur

---

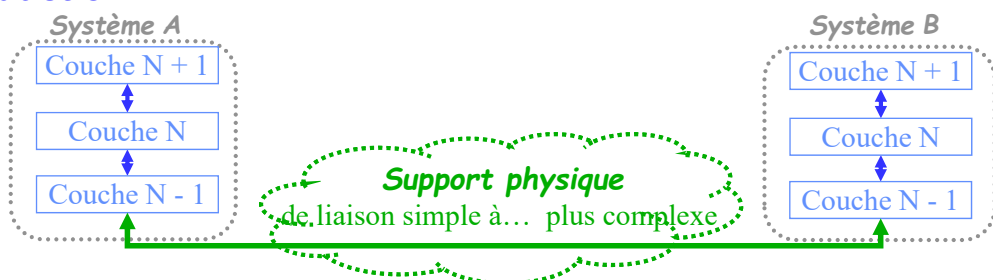
## - 2<sup>ème</sup> partie -

### Modèles Multicouches et d'interconnexion de réseaux ouverts et supports physiques et couches basses

---

## 3 - Modèles multicouches d'interconnexion de systèmes

### ■ 1. Introduction



⇒ Système **OUVERT** = capable, par le respect de règles **normalisées** d'interagir avec d'autres systèmes + d'intégrer de nouveaux "composants"

⇒ Conduit à des architectures multiréseaux de systèmes multicouches (hiérarchisés)

### ■ 2. Besoins et fonctions nécessaires à l'interconnexion

- ⇒ "ÉCHANGER LA SÉMANTIQUE" entre abonnés
- ⇒ "ÉCHANGER LA SYNTAXE" entre abonnés pour utiliser un langage commun
- ⇒ "SYNCHRONISER LE DIALOGUE" entre abonnés
- ⇒ "TRANSPORTER L'INFORMATION" entre abonnés, au besoin via des intermédiaires
  - ④ transférer des données entre applications
  - ⑤ transférer des données entre sites extrêmes : gérer les sites, nœuds intermédiaires
  - ⑥ transférer les données sur entre 2 nœuds
  - ⑦ transférer les bits et les signaux sur le support physique de communication

### ■ 3. Structure multicouche : concepts et modèle de référence de l'ISO



#### 3.1. Couche de niveau N

##### a) Définition

- ⇒ couche N = regroupe Entités/Fonctions travaillant au « **même niveau** » **N d'abstraction**
  - la plus "haute" → gère le **plus haut niveau d'abstraction** : le plus complexe et proche de l'homme
  - la plus "basse" → gère le **plus bas niveau d'abstraction** : bits/signaux pour/sur le support physique

##### b) Interaction inter-couches : une **couche N d'un système A** interagit/échange :

- ⇒ **VIRTUELLEMENT** avec **d'autres couches N** (donc de même niveau d'abstraction) **d'autres systèmes** ⇒ interactions via des **PROTOCOLES**
- ⇒ **RÉELLEMENT** avec **ses couches adjacentes** (donc niveaux d'abstraction différents) **dans le même système A** ⇒ échanges via des **SERVICES** et des « **adresses** » (**SAP : Service Access Point**)

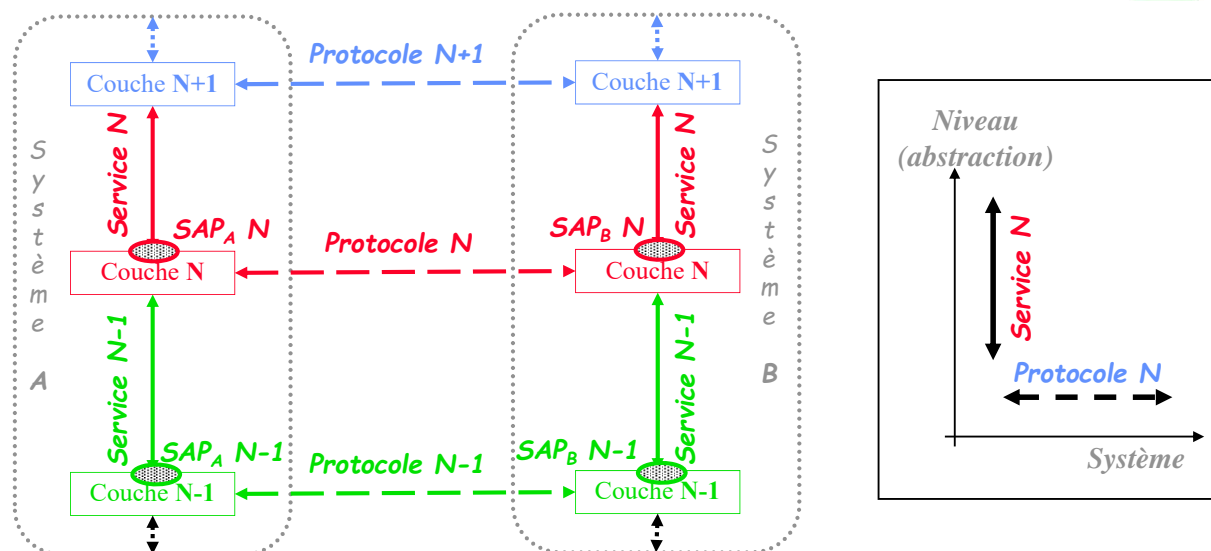
→ une couche N :  
 - **FOURNIT** des services **à sa couche N+1**  
 - **UTILISE** des services **de sa couche N-1**

→ une couche N du système A communique avec la couche N d'un système B, en utilisant des services de sa couche N-1

→ une couche N est caractérisée par :

- ses protocoles (d'échange avec une couche homologue d'un AUTRE système)
- les services qu'elle fournit (à sa couche supérieure dans le MÊME système)

##### c) Schéma



#### 3.2. Entité N (notée E(N) ou E<sub>N</sub>) : une des fonctions assurées par la couche N

#### 3.3. Échanges ... avec sous « **acquiescement** »

Quel que soit le type d'échange (protocole et service), cet échange peut-être défini AVEC ou SANS « **ACQUITTEMENT** » : mécanisme permettant un « retour », positif ou négatif, ou plus complexe que simplement binaire.



### 3.4. Protocole N : caractérise les échanges entre 2 entités de même niveau N d'abstraction (entités « homologues ») de systèmes différents



- ⇒ échanges **VIRTUELS** sous forme de *PDU (Protocol Data Unit)*
- ⇒ 2 grandes classes de PROTOCOLES (avec ou sans **acquittement protocolaire**)
  - ① Orienté connexion → 3 étapes :
    - **Établir la connexion** = établir un lien logique entre 2 entités homologues pour mise en accord avant transfert de données « utiles » : garantir la présence du récepteur ... réserver des "chemins", des buffers, etc.
    - **Transférer des données « utiles »**
    - **Fermer la connexion**
  - ② NON orienté connexion

### 3.5. Service N : caractérise les échanges entre des entités de niveau d'abstraction N et N+1 d'un même système = prestation offerte par une entité de la couche N aux entités de sa couche N+1

#### a) Interactions entre entités ADJACENTES

- ⇒ échanges **RÉELS** sous forme de *SDU (Service Data Unit)*
- ⇒ invoquée par des *primitives de services*
- ⇒ à un instant donné, on a, notamment :
  - une  $E_{N+1}$  peut être en relation avec plusieurs  $E_N$
  - une  $E_N$  peut être en relation avec plusieurs  $E_{N+1}$

#### b) Classes de services, avec ou sans ACQUITTEMENT



- ① Orienté connexion
- ② Non orienté connexion

#### c) Primitives de service de base

- ⇒ 4 types (de base) de primitives servent à construire toutes les primitives de service (attention : tous les services n'utilisent pas toujours les 4 types)
  - ① Demande (**Request**) de prestation/service
  - ② Indication (**Indication**) qui sert à notifier la demande
  - ③ Réponse (**Response**) à l'indication
  - ④ Confirmation (**Confirm**) qui sert à notifier la réponse

⇒ Illustration



#### d) Primitives de services complètes : construites à partir des types de bases

⇒ Format générique d'une primitive de service :

*Niveau\_NomPrimitive\_TypePrimitive { SourceSAP, DestSAP, réponse/SDU ... }*

ex. : **L\_Connect\_Request** {...}, / **L\_DATA\_Response** {...} / **L\_Disconnect\_Request** {...}

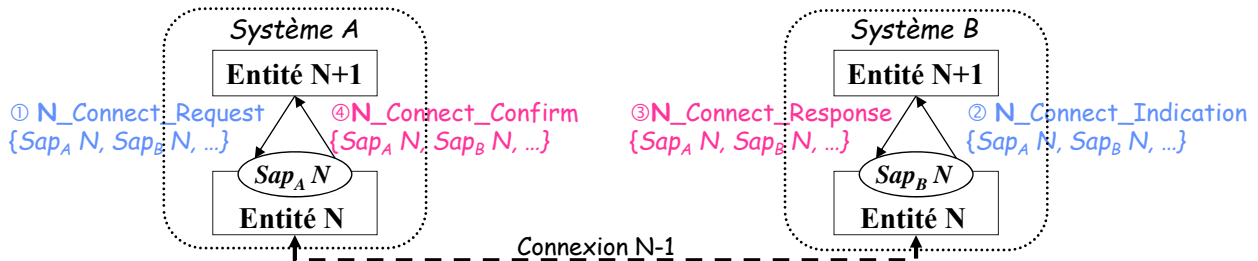
### e) Service DATA UNIT N ( $SDU_N$ )



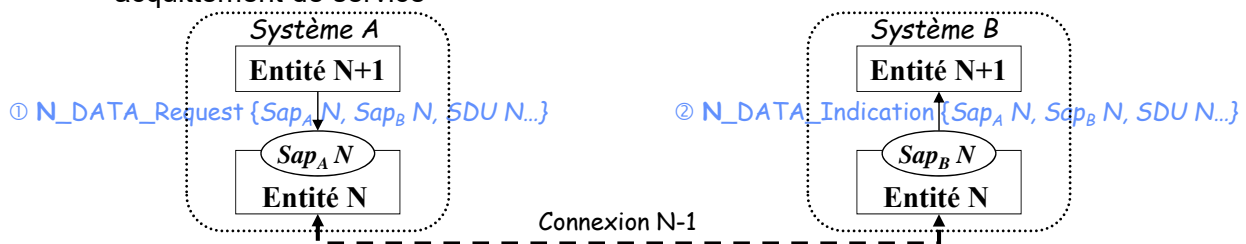
- Unité de "données" échangée entre couche N+1 et couche N
- une  $SDU_N$  sert au transfert réel des informations contenues dans une ou des  $PDU_{N+1}$

### f) Exemples de services

⇒ Service N avec les 4 types de primitives : ex. d'ouverture d'une connexion N+1 avec acquittement de service



⇒ Service N avec 2 types de primitives : ex. de transfert de données de niveau N+1 sans acquittement de service

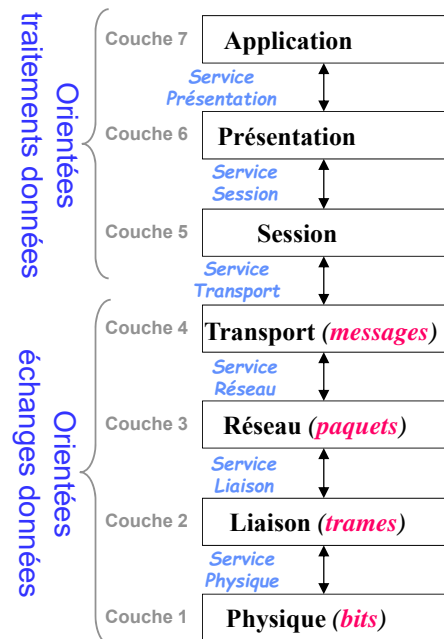


## ■ 4. Modèle OSI de l'ISO

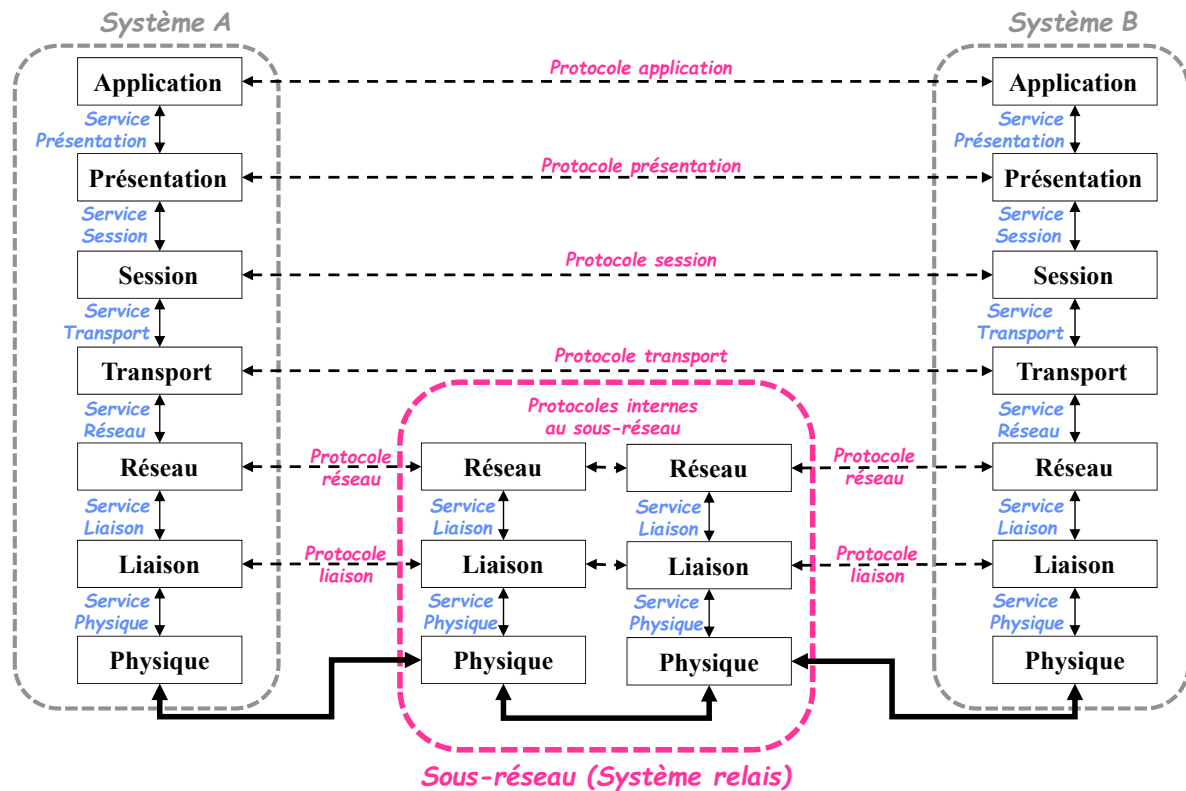


### 4.1. Introduction

- ⇒ OSI = Open Systems Interconnection (1984)
- ⇒ MODÈLE OSI : pas un "produit commercial", mais un "MODÈLE" regroupant définitions et spécifications caractérisant les principales fonctions d'un système de transmission et leurs relations.
  - "guide" de développement de matériels et de logiciels pour l'interconnexion de différents systèmes
  - ne fixe pas les techniques (mécanismes) de mise en oeuvre des fonctions
- ⇒ Constitué de 7 couches/niveaux :
  - l'utilisateur ne voit que la couche 7
  - couches 7 à 5 : dites "hautes", purement logicielles
  - couches 4 et 3 : gère le "**routing**" des données
  - couches 2 et 1 : dites "basses", relatives au matériel
- ⇒ Modèle OSI postérieur à la création d'autres "modèles" réseau, dont TCP/IP, qui continuent quand même à exister



## 4.2. Modèle OSI global



## 4.3. Rôle des différentes couches du Modèle OSI



### ① Couche Physique : gère la transmission des **BITS** sur une ligne/support physique

- transformer "bits ↔ signaux pour un support", c'est à dire transformer
  - bits des trames de la couche 2 → en signaux à émettre
  - signaux reçus → en suites de bits pour la couche 2
- définir pour chaque bit : son encodage, la forme du signal associé et sa durée
- assurer que le niveau d'un bit émis est semblable à la réception
- sens de transmission, récupération d'horloge pour la resynchronisation
- définir le type et brochage des connecteurs, ...

### ② Couche Liaison : gère la communication "entre 2 nœuds adjacents" pour transformer le support de transmission "brut" (couche 1) en une ligne exempte d'erreur à disposition de la couche Réseau ; fournit à la couche 3 une interface unique, indépendante de la nature physique du réseau

- paquets de la couche 3 → trames à émettre, et donc la suite de bits pour le niveau 1
- suites de bits reçues de la couche 1 → constituer les trames reçues
- **gérer l'accès d'un nœud au support de communication**
- **contrôler les erreurs sur les bits, et les corriger si possible + contrôler le flux des trames** (pertes et duplications de trames)

③ Couche Réseau : gestion de « bout en bout » entre machines-hôte → gère toutes les fonctions de « relais »

- les nœuds intermédiaires sont encore/déjà visibles
- Router/commuter les paquets : déterminer la route à suivre par les paquets entre l'émetteur et le récepteur en passant par des nœuds intermédiaires.
- Contrôler la congestion d'un routeur, contourner des "éléments" en panne
- passer d'un type d'adressage à un autre (interconnexion adressages hétérogènes)
- Couche quasi inexistante dans les cas de : transmission par diffusion, ou de machines directement interconnectées

④ Couche Transport : gestion de « bout en bout » entre processus (en cours d'exécution sur les machines-hôtes) ; fournit à la couche Session, un transport avec une certaine *qualité de service* (Quality Of Service - QoS)

- communication "directe" entre les processus (sans considération de nœuds intermédiaires)
- « raisonne » sur les messages dans leur intégralité (et non pas sur des portions)
- fournit à la couche Session, un transport avec une certaine qualité de service (Quality Of Service - QoS)
- si besoins de débits élevés, création de plusieurs « connexions réseau »

⑤ Couche Session : permettre des sessions entre UTILISATEURS

- établir une entente entre **utilisateur-émetteur** et **utilisateur-récepteur** (ouverture de connexion), transférer des données, terminer la session (fermeture de connexion);
- services avancés pour certaines applications (ex. : synchronisation en positionnant des points de repères dans les données, ...)

⑥ Couche Présentation : gère la « représentation (codage) » des données → préparer les données pour la "transmission" et présenter à la couche 7 des données dans un format unique par rapport à ceux possibles pour la couche 6

⑦ Couche Application : interface avec l'utilisateur

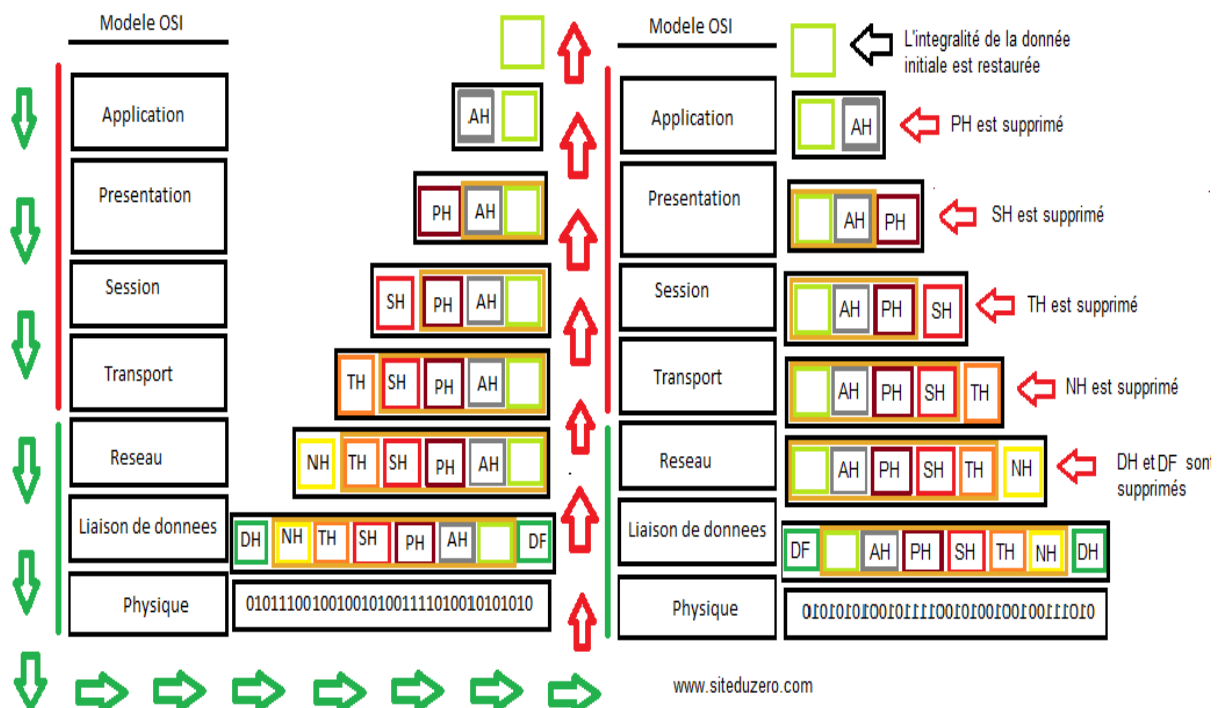
- émulation de terminal virtuel : rendre compatible terminal virtuel et terminal réel
  - ISO : *VTS* (Virtual Terminal Service)
  - Internet : *telnet* (en mode texte)
- soumission de travaux à distance :
  - ISO : *JTM* (Job Transfert Management)
  - Internet : *rlogin*, *ssh*
- transfert de fichier,, ...
  - ISO : *FTAM* (File Transfert Access and Management)
  - Internet : *ftp* (File Transfert Protocol), *http* (Hyper Text Transfert Protocol)
- messagerie électronique
  - ISO : *10021*
  - Internet : *smtp* (Simple Mail Transfert Protocol) et *mail*



### 5.1. Définitions

- ⇒ Une couche N « reçoit », de sa couche supérieure N+1, des unités d'informations « utiles », qu'elle doit transférer vers une couche homologue via un protocole. Des informations spécifiques au protocole donné, sont ajoutées avant et après ces des unités d'informations « utiles »
- ⇒ Les unités d'informations "utiles" sont donc **ENCAPSULÉES**. La couche N réceptrice dans l'autre système, devra « **DÉSENCAPSULER** » avant d'envoyer à la couche N+1
- ⇒ Selon la couche N, son protocole et la taille des unités de données venant de la couche supérieure, la couche N peut être amenée à découper ces unités en morceaux, avant d'encapsuler chaque morceau, ou, autre cas, à regrouper plusieurs unités.
- ⇒  $PDU_N =$ 
  - soit  $PCI_N$  (Protocol Control Information) uniquement
  - soit  $PCI_N + SDU_N$
  - soit  $PCI_N + FSDU_N$  (Fragment de  $SDU_N$ )
- ⇒  $SDU_{N-1} =$  selon la taille des  $PDU_N$ , à :
  - une partie d'un seul  $PDU_N$
  - tout un  $PDU_N$  dans sa globalité
  - plusieurs  $PDU_N$  dans leur globalité

### 5.2. Exemple dans le modèle OSI





## ■ 6. Adaptations du modèle OSI et Autres modèles multicouches

### 6.1. Pourquoi des adaptations et d'autres modèle ?

- ⇒ OSI = un modèle d'architecture multicouche "générique", visant à couvrir un maximum de cas ⇒ pas suffisamment adapté, voire détaillé pour certains cas
- ⇒ OSI = un modèle d'architecture multicouche parmi d'autres, auxquels il est :
  - postérieur → il en intègre certains aspects, mais ces modèles perdurent (ex : Internet)
  - antérieur → ils s'y réfèrent tout en s'en écartant pour offrir des améliorations et être mieux adaptés à certains besoins

### 6.2. Un exemple d'adaptation du modèle OSI : les réseaux locaux

#### a) Spécificités des Réseaux locaux et des réseaux locaux industriels (RLI)

- ⇒ "**LOCAL**" : la distance initialement, mais plus seulement maintenant (autres aspects vus après)
- ⇒ "**INDUSTRIEL**" : systèmes de commande de procédés critiques, avec répartition du matériel (calculateur, capteurs et actionneurs), voire des fonctions de commande, et donc échanges de commandes, de calculs, d'alarmes ⇒ fortes contraintes de
  - **temps réel** : besoin d'une borne supérieure pour le temps de transmission
  - **sûreté et sécurité** : disponibilité et tolérance aux fautes (pannes, intrusions)

Problème crucial : l'accès au réseau pour chaque machine ⇒ principale spécificité se situe au niveau du type de "protocole liaison" :

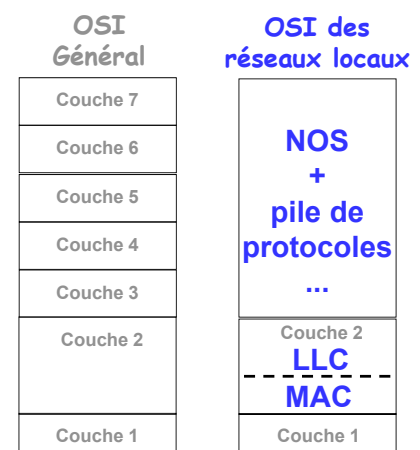
- taux d'erreurs plus faible, débits élevés
- ex : Ethernet, puis token ring d'IBM, CAN, ...

#### b) Couche 2 : rôle prépondérant pour la gestion de l'accès au support ⇒ découpée en 2 sous couches

- ⇒ **MAC** (Medium Access Control) : mécanismes et politiques d'accès au support (ex : CSMA)
- ⇒ **LLC** (Logical Link Control) : mécanismes d'établissement de liens logiques ⇒ accomplit les fonctions habituelles de la couche 2 (contrôle d'erreur, de flux, ...)

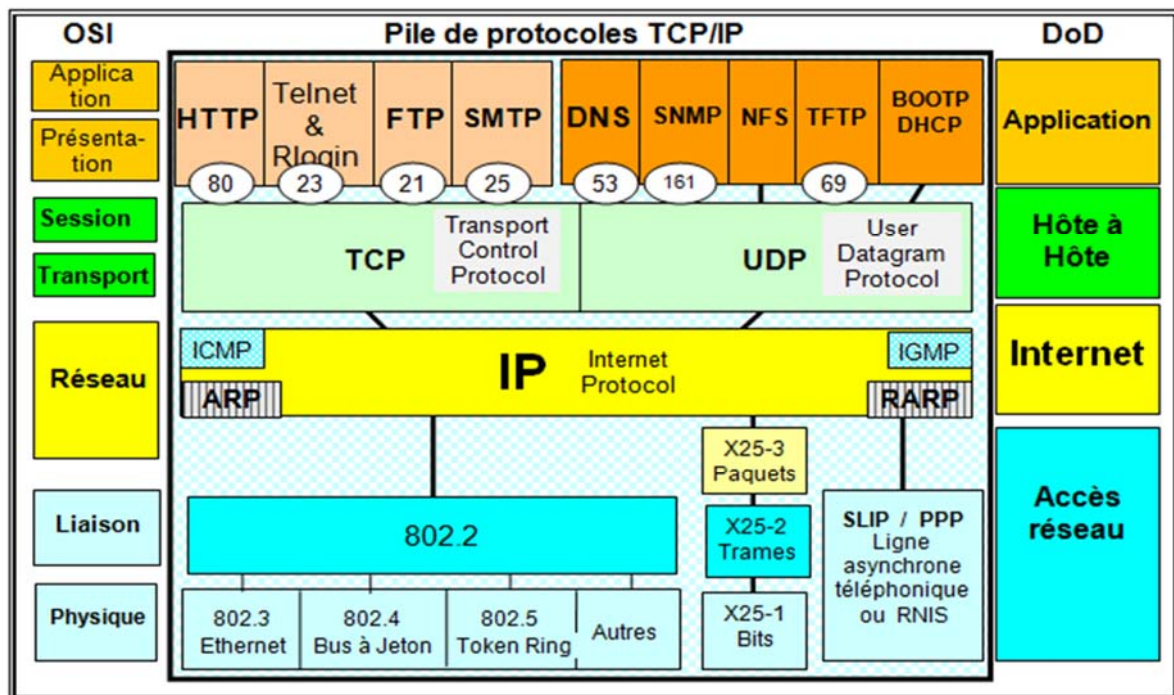
#### c) Couches 3 à 7

- ⇒ Traitements dans ces couches réduits ⇒ ne justifient plus 5 couches (ex. : routage de niveau 3 souvent inutile) ⇒ ces traitements sont regroupés en une seule "couche", correspondant au NOS
- ⇒ **NOS** (Network Operating System) : prise en charge du réseau par les applications, en s'appuyant sur une pile de protocoles (souvent représentés inclus dans le NOS)
- ⇒ **Ex. de NOS** : Appeltalk, PCSA de DEC, Netware de NOVELL, LAN server d'IBM, Window NT, NFS d'UNIX
- ⇒ **Ex. de protocoles** : MAP de l'ISO, TCP/IP du monde UNIX, IPX/SPX de NOVELL, NETBIOS d'IBM, NETBUI de Windows



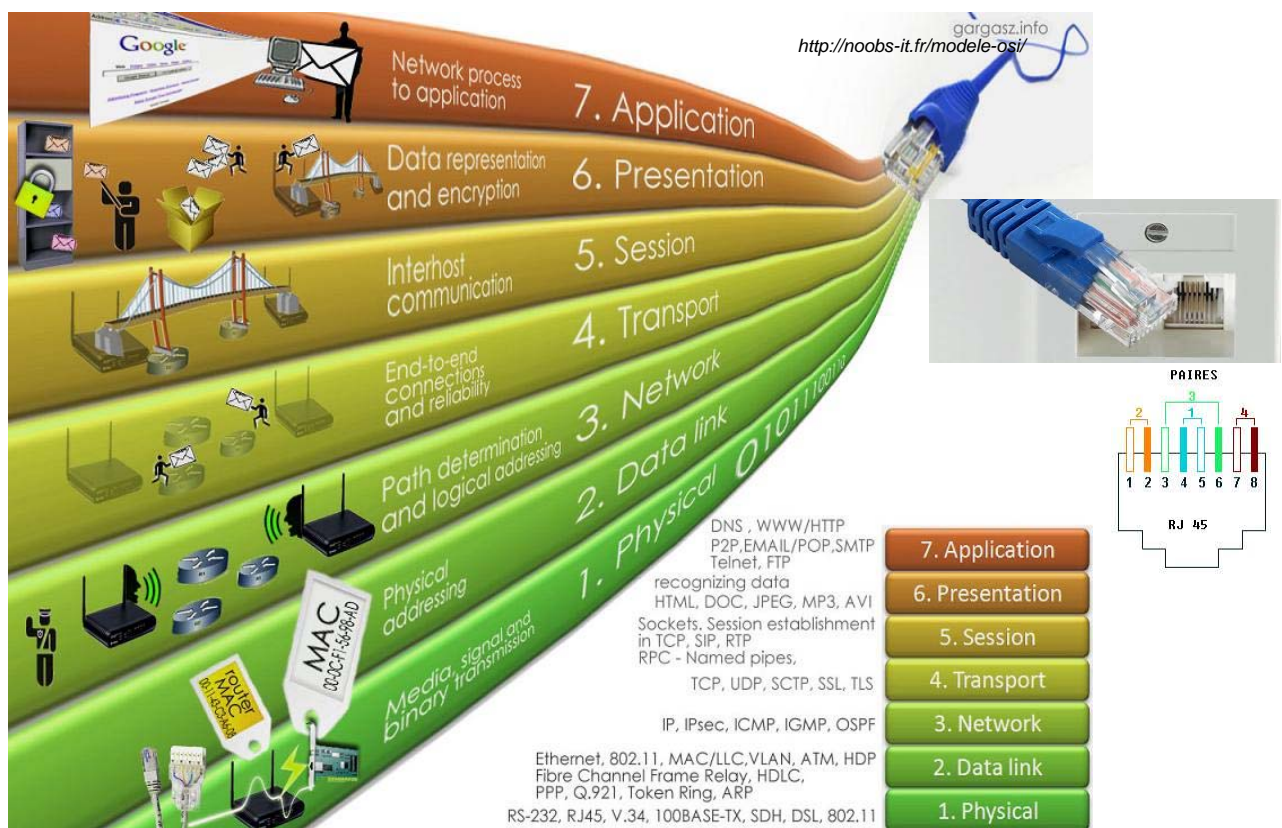
### 6.3. Autres modèles multichouches

#### a) Modèle DoD et « piles de protocoles TCP/IP »



#### b) Autres modèles multichouches ... la totale : [https://fr.wikipedia.org/wiki/Mod%C3%A8le\\_OSI](https://fr.wikipedia.org/wiki/Mod%C3%A8le_OSI)

#### c) autre figure ... attention : des effets visuels trompeurs



## ■ 7. Sous-réseaux : interconnexions de réseaux



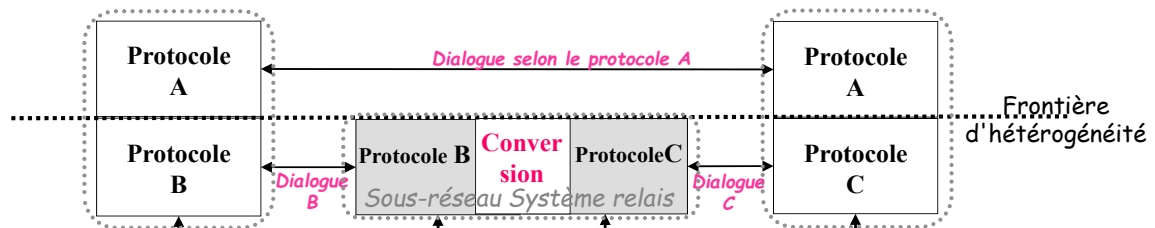
### 7.1. Introduction

⇒ Notamment avec les Réseaux locaux hétérogènes, est apparu le besoin plus général d'interconnecter des réseaux différents. Cette interconnexion peut se faire :

- par des appareils dédiés, nommés "relais"
- par réseau, dit de transport (ex : X25)

⇒ Notion de relais : 2 réseaux à interconnecter peuvent utiliser des protocoles différents ⇒ un relais doit assurer 3 techniques d'interconnexions :

- ① la conversion de service
- ② la conversion de protocole
- ③ l'encapsulation



⇒ Les 4 types de relais (normalisés ISO), selon les couches

- couche 1 : **RÉPÉTEURS** (éléments **passifs** du réseau)
- couche 2 : **PONTS** (éléments **actifs** du réseau)
- couche 3 : **ROUTEURS** (éléments **actifs** du réseau)
- couche 4 et plus : **PASSERELLES** (éléments **actifs** du réseau)

### 7.2. Répéteur : agit au niveau physique



⇒ Interconnexion de brins "physiques" d'un réseau

⇒ Principe : répète (régénère) en sortie tous les signaux reçus en entrée,

- sans aucun discernement ⇒ aucun effet sur le contenu logique ni sur le protocole
- il n'a pas d'adresse (n'est pas adressable)

⇒ Utilisations :

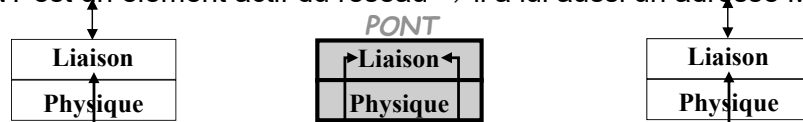
- ① **régénération du signal** : s'affranchir de la limite de distance d'un support (ex : brin de 200 m en 10 base 2)
- ② **adaptation de support** (ex : passage de coaxial en fibre optique)
- ③ **isolation galvanique** (électrique) entre sous-réseaux

### 7.3. Pont

⇒ Élément d'interconnexion de "brins" dans un réseau au niveau de la sous-couche MAC

⇒ espace unique d'adressage de niveau 2 du réseau : les adresses MAC

⇒ Un PONT est un élément actif du réseau ⇒ il a lui aussi une adresse MAC

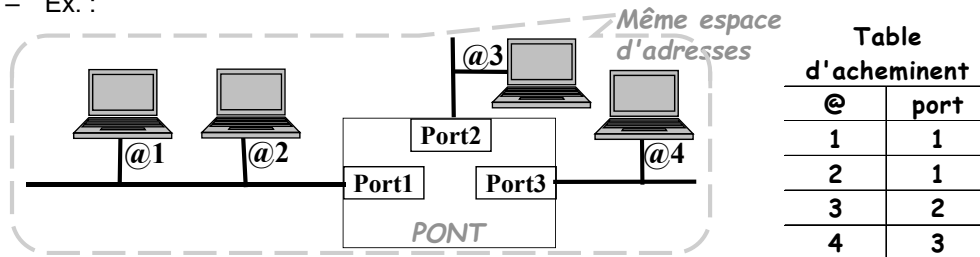


## a) Fonctionnalités temps réel d'un pont



- ⇒ **Adaptation** de débits ou de support
- ⇒ **"Routing" de niveau 2** car peut interpréter les trames pour connaître l'adresse MAC du destinataire ⇒ un Pont peut contenir une table d'acheminement.

– Ex. :



- Changer une machine de localisation physique dans le sous-réseau est invisible pour les autres machines.
  - Routage partiel au niveau 2 ⇒ un pont n'est pas totalement conforme à la norme OSI
- ⇒ **Filtrage** ⇒ limiter le trafic dans le réseau local
- **"Isolation"** de sous-réseaux. Sur l'exemple, le brin avec @1 et @2 est partiellement isolé du brin avec @3 et de celui avec @4 : le pont ne diffuse pas tous les messages d'un brin vers tous les autres "brins", mais que les messages d'un brin concernant un autre brin
  - **contrôle d'erreurs** : ne transmet pas les trames incorrectes (en nombre ou valeur de bits)

## b) Architectures basées sur des Ponts

- ⇒ Les architectures de réseaux habituellement construites en utilisant des ponts sont dites "BACKBONE" : un brin du réseau fédère les autres brins s'y rattachant

## c) Les différents Ponts : classés selon 2 critères

- la manière dont est construite leur table d'acheminement
  - la distance qui sépare les réseaux interconnectés
- ① **Ponts Simples** : trames acheminées selon une **table d'acheminement statique, construite manuellement** à la configuration du pont.
- ② **Ponts transparents (à apprentissage)** : très courant dans les réseaux Ethernet
- ⇒ Table d'acheminement **auto construite dynamiquement par apprentissage**
  - ⇒ Apprentissage effectué par analyse des @MAC émettrices de trames
    - si une destination n'est pas connue, le pont retransmet la trame sur tous ses ports de sortie (sauf celui par lequel la trame est arrivée)
    - pour éviter d'avoir des tables grossissant indéfiniment, les @ les plus anciennes sont périodiquement effacées de la table
  - ⇒ PB de bouclage : la mise en parallèle de 2 ponts peut provoquer un phénomène de bouclage ⇒ l'IEEE a normalisé un protocole d'apprentissage permettant de mettre en sommeil les ponts redondants : protocole IEEE 802.1D, appelé "SAPNNING TREE Protocol"

③ **Ponts à routage par la source (Source Routing)** : surtout dans réseaux Token Ring

⇒ pas de table d'acheminement dans le pont : elles sont directement maintenues dans les stations sur le réseau ⇒ le chemin à suivre est placé dans les trames.

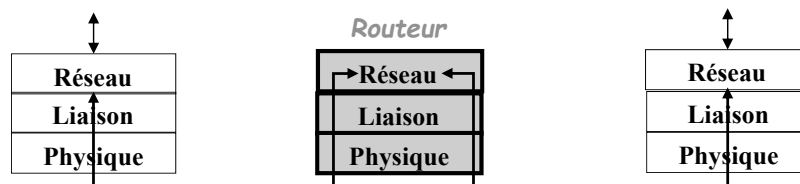
④ **Ponts SRT** : surtout pour les milieux hétérogènes

⇒ A la fois Source routing et Transparent

## 7.4. Routeurs

⇒ Élément d'interconnexion de sous-réseaux au niveau de la couche 3

⇒ les sous-réseaux peuvent avoir des **espaces d'adressage différents**



⇒ Achemine des données vers un destinataire défini par une @ de niveau 3 (ex. : IP)

⇒ conversion de protocoles

## 7.5. Passerelles

⇒ Éléments d'interconnexion d'entités de niveau 4 et plus, ayant des **espaces d'adressage homogènes**

⇒ Remarque : il existe d'autres définitions



## 7.6. Autres matériels d'interconnexion (très utilisé dans les réseaux locaux)

① **HUB** : interconnexion d'entités de **niveau « 1 »**

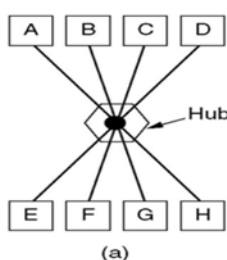
⇒ Régénère et duplique TOUTES les trames qu'il reçoit en entrée sur TOUS ses ports : un HUB ne fait que de la diffusion : débit du HUB divisé par le nombre de port

⇒ Permet d'implémenter une **topologie logique de type bus**, sur la base d'une **topologie physique de type étoile**

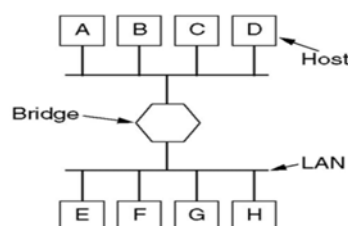
② **COMMUTATEUR** : interconnexion d'entités de **niveau « 2 » (et donc 1 aussi)**

⇒ Fonctions « similaire » à celui d'un PONT (possibilité de « router » les trames que vers les destinataires) : débit « max » attribuable à chacun de ses ports

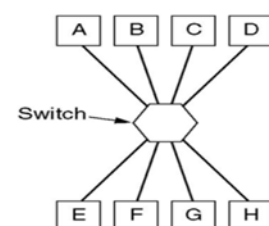
⇒ Topologie « similaire » à celle d'un HUB.



(a)



(b)



(c)