



# Introduction aux réseaux

UTSEUS – UESR03

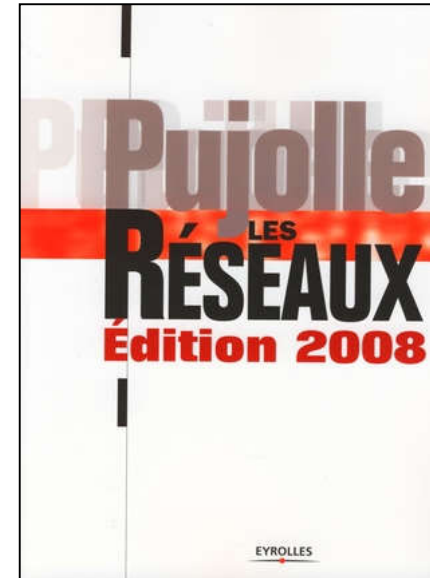
Moez ESSEGHIR

[Moez.esseghir@utt.fr](mailto:Moez.esseghir@utt.fr)

# Références

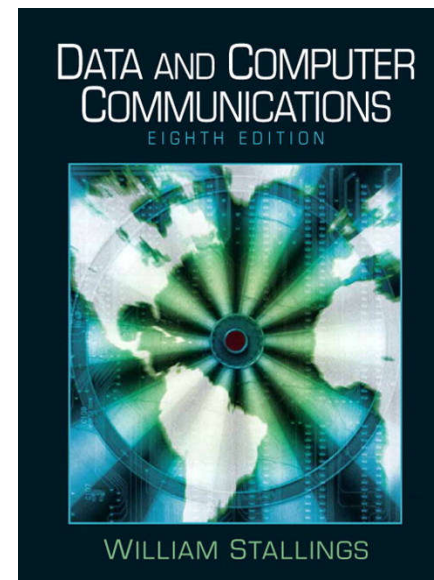
## 1. Les réseaux

- Auteur : Guy Pujolle
- Editeur : Eyrolles
- Langue : Français



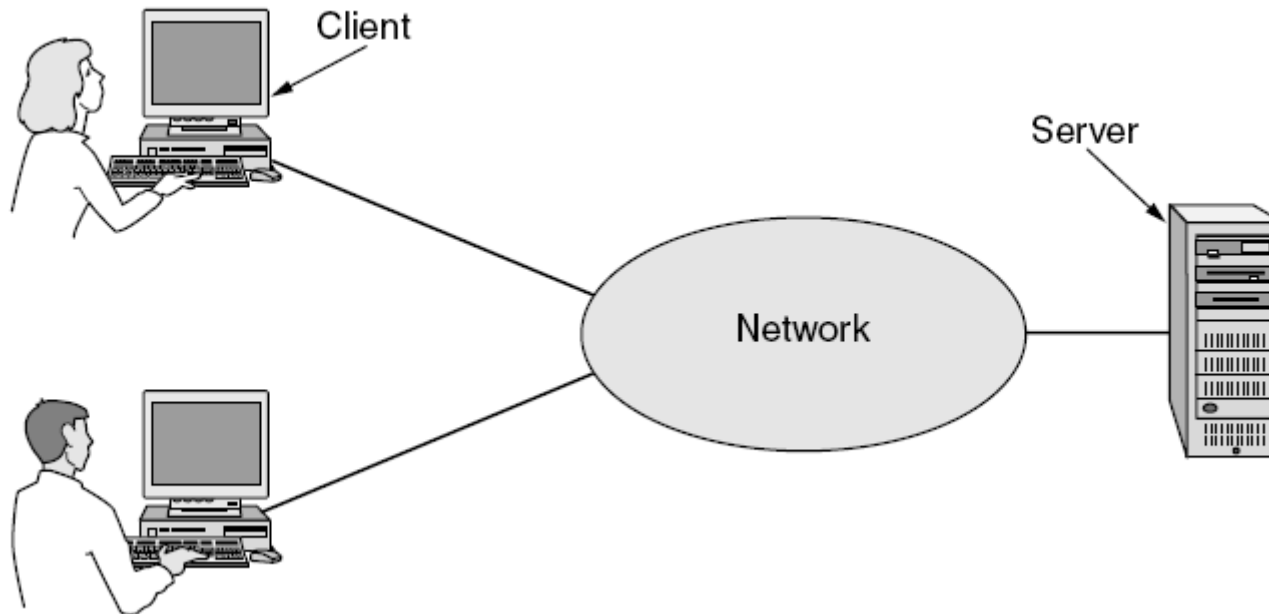
## 2. Data and Computer Communication

- Auteur : William Stallings
- Editeur : Prentice Hall
- Langue : Anglais



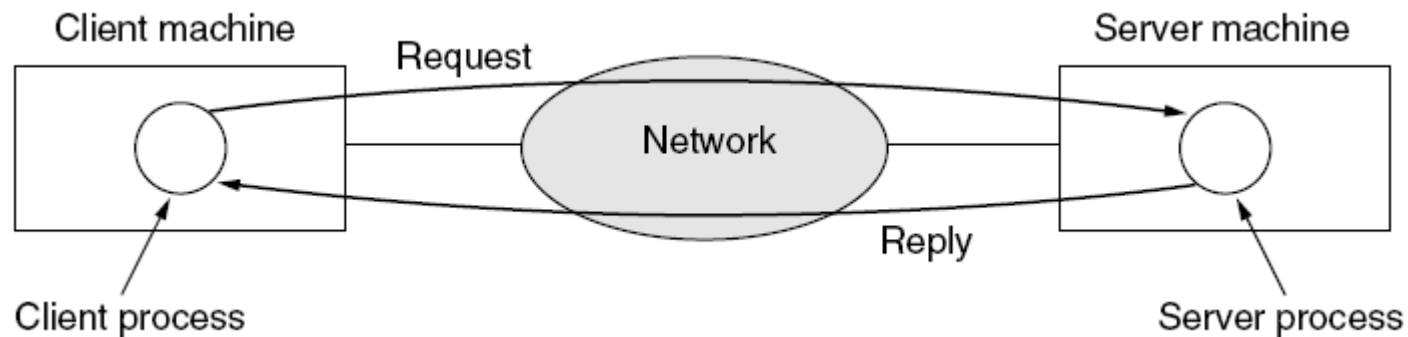
# Utilisation des réseaux

- Applications professionnelles
  - Partage de ressources : applications, équipements, données visibles sur un réseau indépendamment de leurs localisations physiques
  - **VPN** (Virtual Path Networks) pour interconnecter des sites différents
  - Modèle **Client - Serveur**



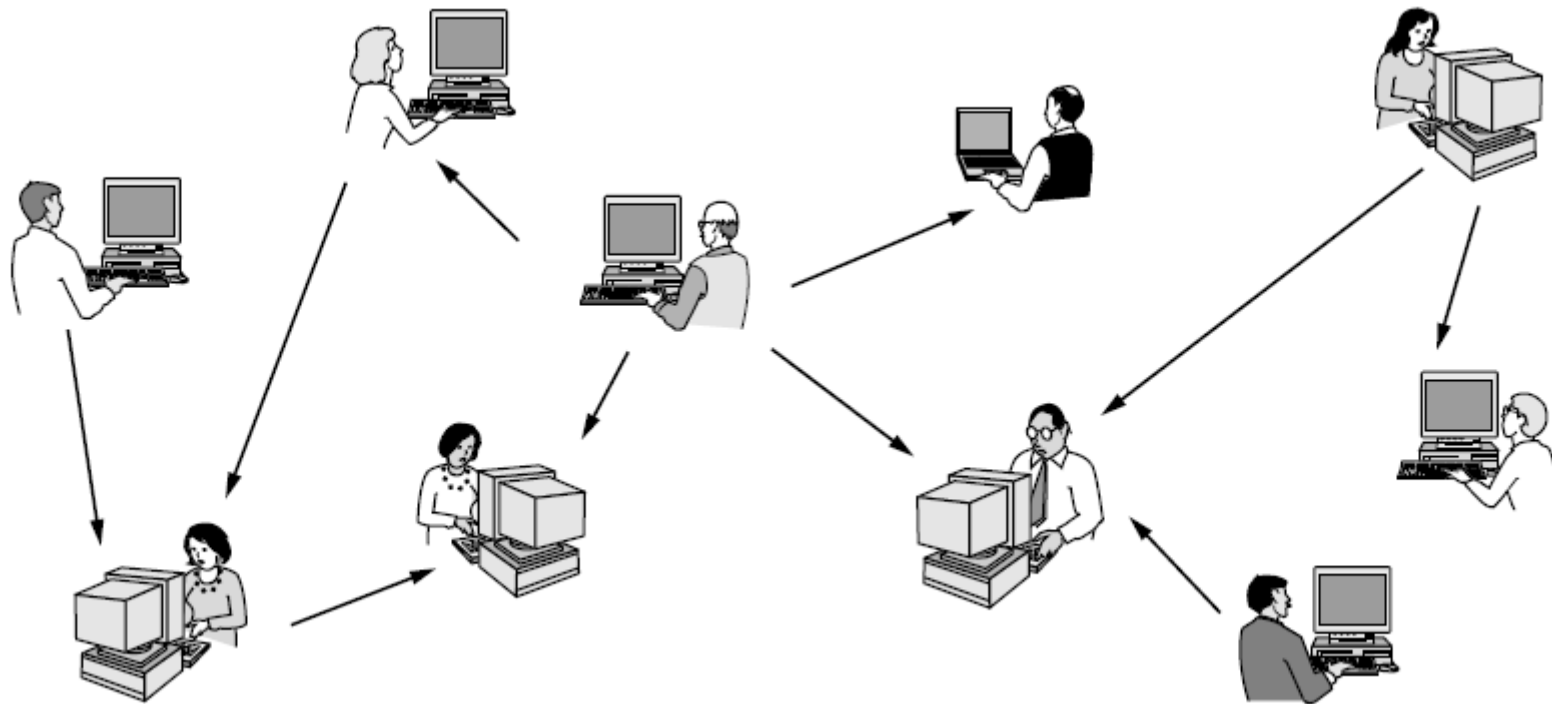
# Utilisation des réseaux

- Applications professionnelles
  - Partage de ressources : applications, équipements, données visibles sur un réseau indépendamment de leurs localisations physiques
  - **VPN** (Virtual Path Networks) pour interconnecter des sites différents
  - Modèle **Client - Serveur**

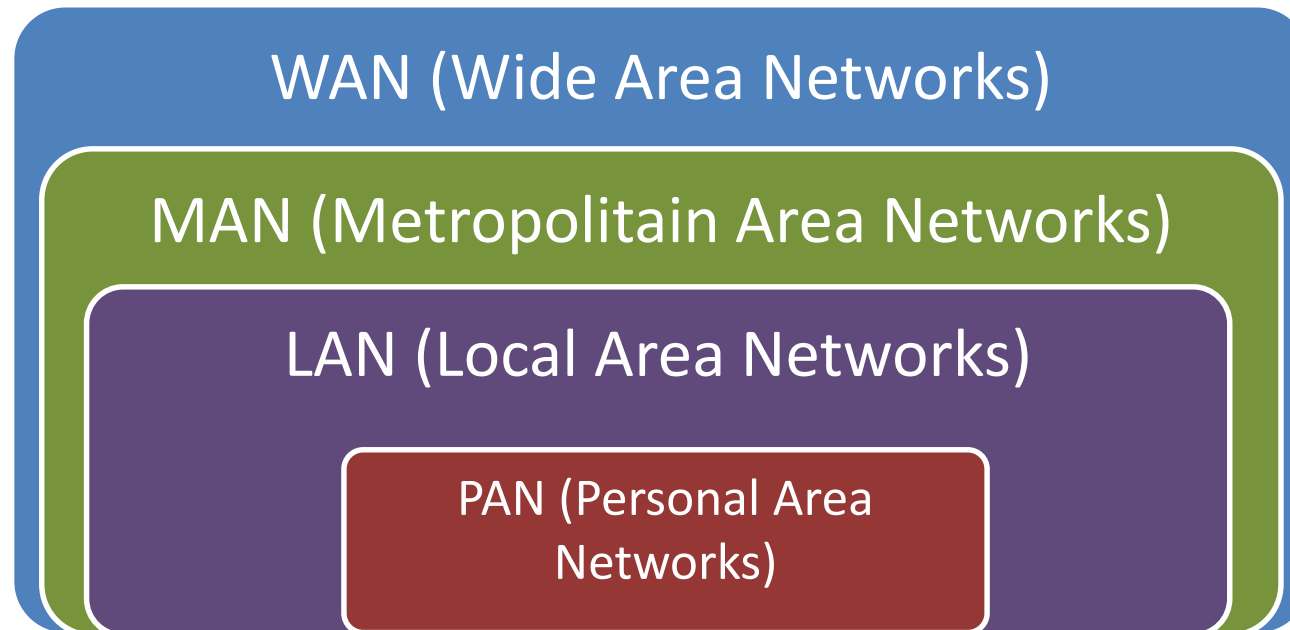


# Utilisation des réseaux

- Applications personnelles 万恶的P2P模式
  - Modèle **pair à pair**:



# Classification des réseaux



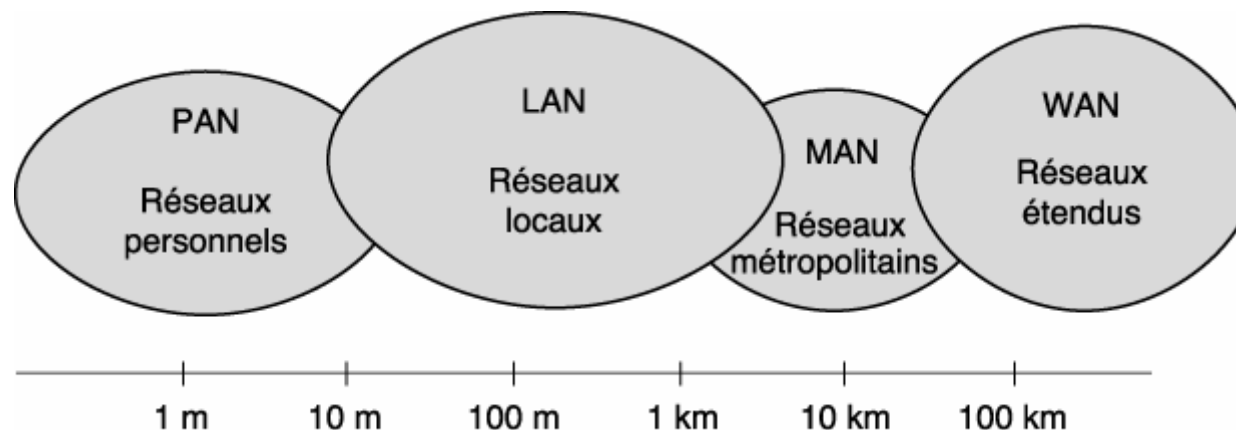
- Suivant la taille géographique on distingue 3 classes de réseaux:
  - Les réseaux **étendus** (WAN)
  - Les réseaux **métropolitains** (MAN)
  - Les réseaux **locaux** (LAN)
  - Les réseaux **personnels** (PAN)

# Classification des réseaux

- Les réseaux WAN:
  - A faible ou moyen débit: réseaux téléphoniques, réseaux de transmission de données -低速或中速: 电话网络、数据传输网络
  - A grand débit: réseaux câblo-opérateurs, réseaux satellites
- Les réseaux LAN:
  - A courte distance et grand débit
  - Parmi les réseaux locaux on distingue les **réseaux départementaux DAN** (*Departmental Area Network*) et les réseaux d'établissement
  - Un réseau **DAN** appelé aussi **réseau capillaire** a pour objectif de relier les réseaux d'un même département situé souvent dans un même étage
  - Un réseau d'établissement relie des réseaux **DAN**
  - Un **DAN** peut être confiné dans un seul bâtiment: **BAN** (*Building Area Network*) ou desservir plusieurs bâtiments géographiquement très proche: **CAN** (*Campus Area Network*)

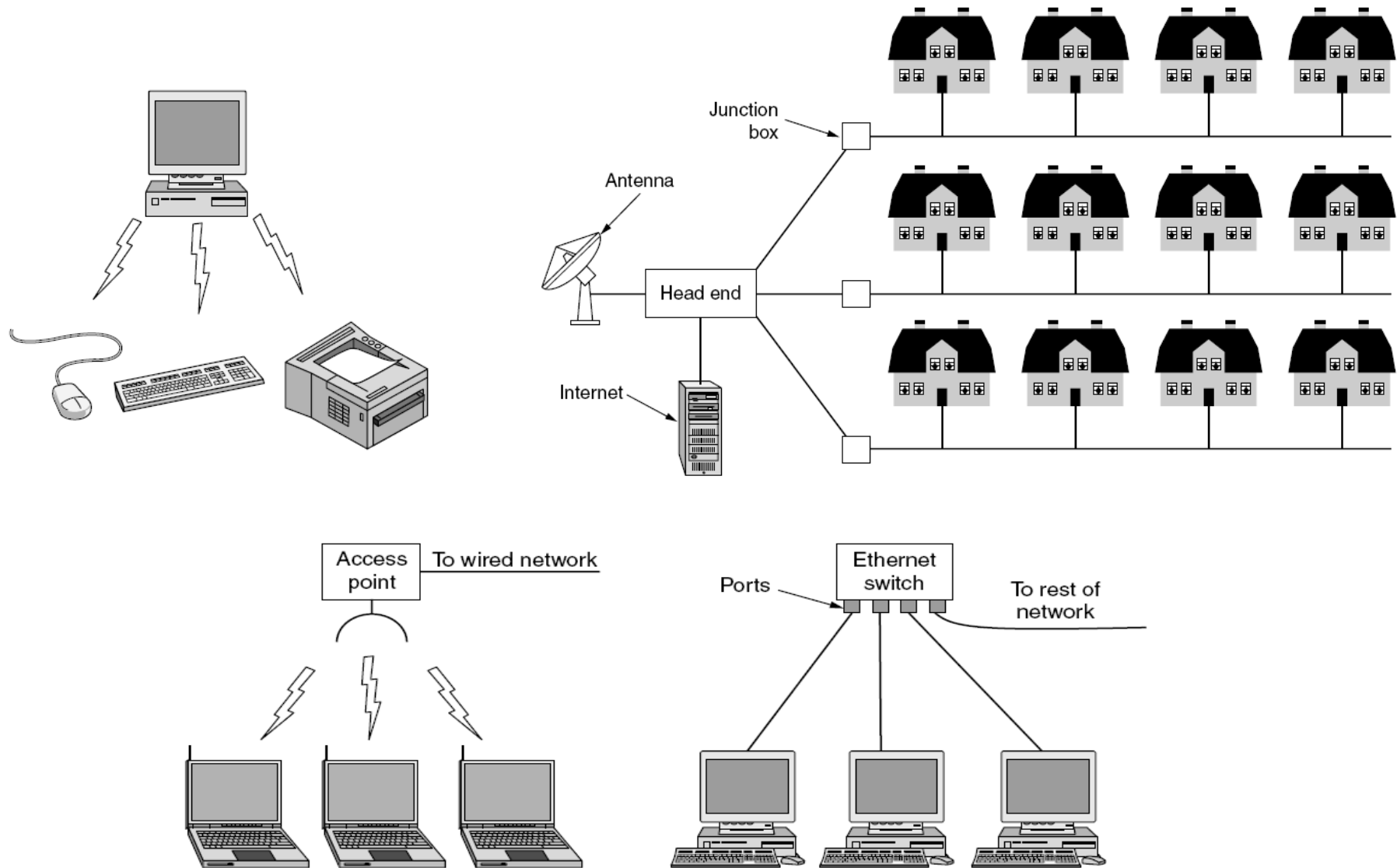
# Classification des réseaux

- Les réseaux MAN:
  - Une extension des réseaux locaux est celle où les bâtiments, d'une même entreprise, peuvent être distants de plusieurs dizaines de kilomètres dans une même métropole (<100km)
  - Les communications se font alors par l'intermédiaire d'une infrastructure de télécommunication (*carrier*)





# Classification des réseaux

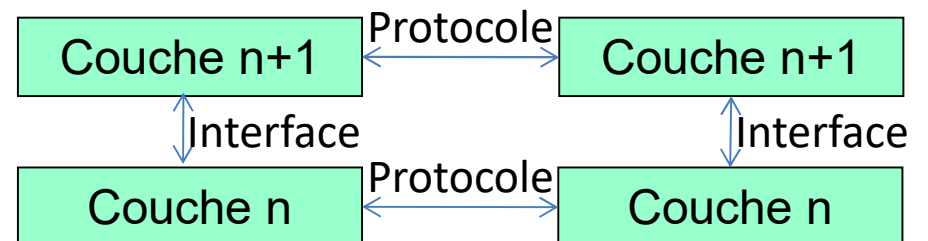


# Modèle: Objectifs

- Dans les réseaux, on s'intéresse aussi bien à l'aspect matériel (câblages, équipements, etc.) qu'à l'aspect logiciel
- Le logiciel dans le réseau est très *structuré*: Pour réduire la complexité de conception, la plupart des réseaux sont organisés en **couches** ou **niveaux**
  - Chaque couche est construite sur la précédente
  - Le nombre de couches, leur nom, leurs fonctions varient selon le réseau
  - L'objet de chaque couche est d'offrir certains services aux couches plus hautes, sans qu'elles aient à connaître les détails d'implémentation de ces services

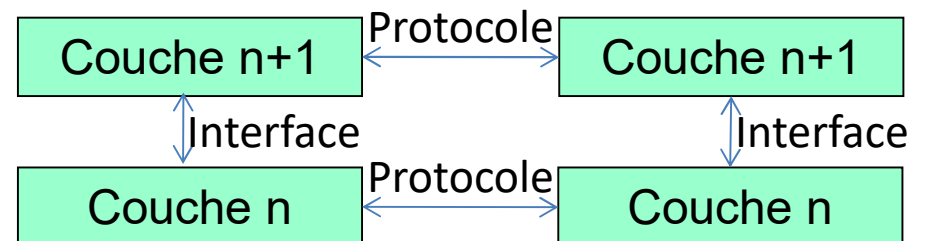
# Modèle: Objectifs

- La couche de niveau  $n$  d'une machine gère la conversation avec la couche  $n$  d'une autre machine
- Les règles et conventions utilisées pour cette conversation constituent le **protocole** de la couche  $n$
- Un protocole c'est un peu un accord entre les parties sur la façon de communiquer
- Toute violation du protocole rend la communication difficile voire même impossible
- Les entités comprenant les couches correspondantes sur différentes machines sont appelés **processus pairs**: Ils communiquent en utilisant le protocole



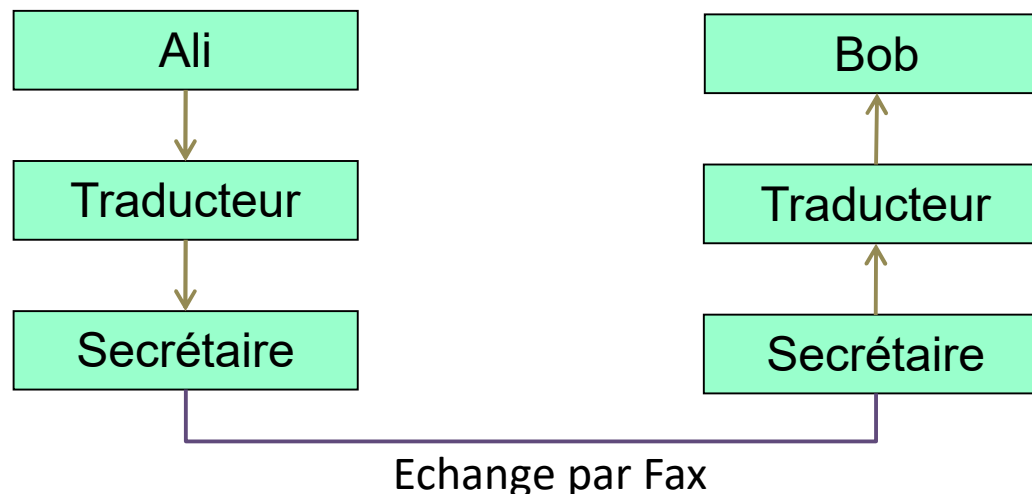
# Modèle: Objectifs

- En réalité, aucune donnée ne passe directement de la couche  $n$  d'une machine à la couche  $n$  d'une autre machine
- Chaque couche passe les données à la couche immédiatement inférieure jusqu'à la plus basse
- En dessous de celle-ci se trouve **la couche physique** qui véhicule réellement la communication
- Entre chaque paire de couches adjacentes, il y a une **interface**
- L'interface définit les opérations élémentaires et les services que la couche inférieure offre à la couche supérieure
- Chaque couche réalise un ensemble de fonctions bien définies



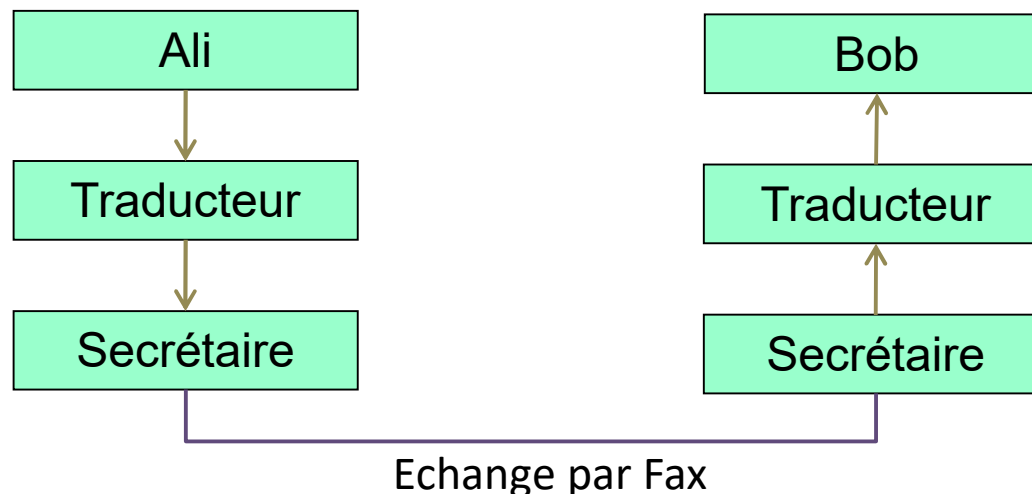
# Modèle: Objectifs

- Des interfaces bien conçues facilitent les changements d'implémentation dans une couche
- L'ensemble des couches et protocoles constitue **l'architecture du réseau**
- L'ensemble des protocoles utilisés par un système, avec un protocole par couche, est appelé **pile de protocoles**
- Exemple: Ali parle anglais et arabe. Bob parle chinois et français.
  - Pas de langue commune. Ils engagent chacun un traducteur. Les traducteurs contactant à leur tour un secrétaire



# Modèle: Objectifs

- Ali passe son message en utilisant l'interface avec le traducteur: l'anglais
- Les traducteurs se sont mis d'accord sur une langue pivot, l'allemand. C'est le protocole de couche 2 entre les processus pairs de niveau 2
- Le traducteur donne alors le message à son secrétaire pour transmission par fax (protocole de niveau 1)
- Quand le message arrive, il est traduit en français et transmis vers Bob.
- Notez que les protocoles sont complètement indépendants tant que l'interface reste inchangée. Les traducteurs peuvent changer de langues sans rien changer sur les interfaces.

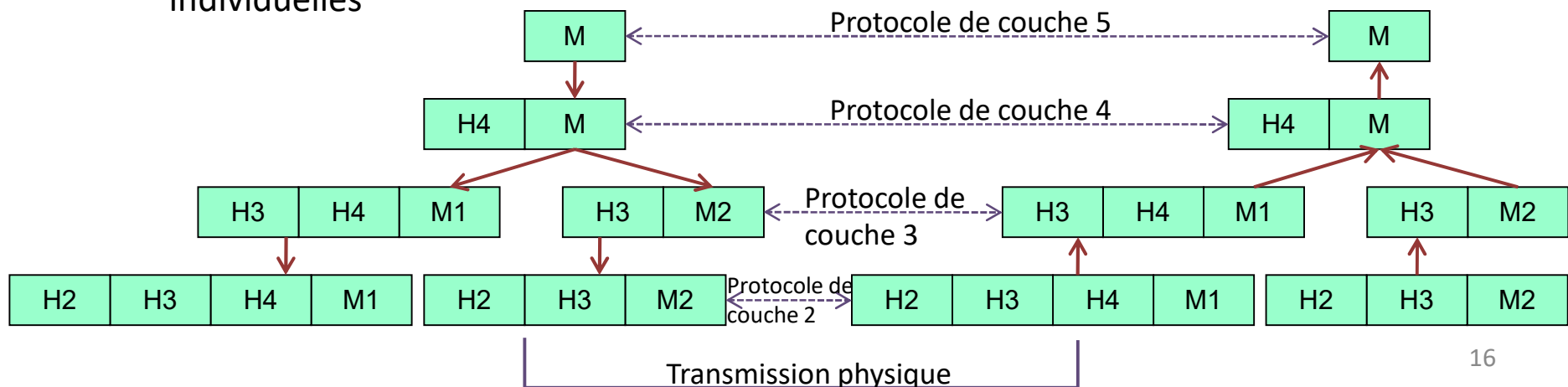


# Modèle: Objectifs

- Prenons maintenant un exemple plus technique sur un réseau à 5 couches
- Un **message** est produit par une application s'exécutant sur la couche 5 et passé à la couche 4 pour transmission
- La couche 4 place un **en-tête** au début du message pour l'identifier et passe le résultat à la couche 3
- Cet en-tête comporte *de l'information de contrôle*, un numéro de séquence par exemple, qui permettra à la couche 4 de la machine destination de remettre le message dans le bon ordre même si les couches inférieures ne savent pas garantir l'ordre des messages
- Les en-têtes peuvent contenir des tailles, des temps limites et autres valeurs
- Dans notre cas, il n'y a pas de limite à la taille des messages acceptés par la couche 4, mais une limite est imposée par la couche 3
- La couche 3 doit alors fractionner les messages entrants en unités plus petites, auxquelles elle ajoute un en-tête

# Modèle: Objectifs

- Dans la machine de réception, les messages montent, de couche en couche, les en-têtes étant supprimés durant la progression
- Aucun en-tête d'une couche inférieure à  $n$  n'est passé à la couche  $n$
- Les protocoles pairs de couche 4 conçoivent leur communication de façon horizontale, grâce au protocole de couche 4
  - Chacun est censé avoir une procédure « envoyer à l'autre côté » et une procédure « réception de l'autre côté », même si ces procédures communiquent avec les couches inférieures via l'interface 3/4
- Le concept abstrait de processus pair est crucial pour la conception de réseaux
  - Sans cette technique, il est difficile de passer d'un problème insoluble à plusieurs problèmes appréhendables, en l'occurrence à la conception de couches individuelles





# Modèle: Principes

- Les principes clés de conception de réseau sont présents dans plusieurs couches
- Chaque couche doit posséder un mécanisme d'identification des émetteurs et des récepteurs
- Chaque machine ayant également plusieurs processus, il doit y avoir un moyen approprié permettant à un processus de spécifier avec qui il veut établir une connexion
  - Il est nécessaire d'avoir *une forme d'adressage* particulière pour préciser chaque destination
- Les contrôles d'erreur est un problème important parce que les supports physiques de transmission ne sont pas parfaits
  - Beaucoup de codes de détection et de correction d'erreurs existent mais les deux extrémités de la connexion doivent se mettre d'accord sur celui qu'elles utiliseront
  - Le récepteur doit disposer d'un moyen d'indiquer à l'émetteur quels messages ont été correctement reçus

# Modèle: Principes

- Tous les canaux de communication ne préservent pas l'ordre des messages qu'ils acheminent
- Etant donné la possibilité d'une perte de séquence, le protocole doit fournir au récepteur les moyens de remettre les paquets dans le bon ordre
  - Recours à la numérotation
- Un autre problème qui se pose à tous les niveaux est qu'un *émetteur rapide* peut saturer de données un récepteur lent
- Il y a également l'incapacité de tous les processus à accepter des messages d'une longueur quelconque
  - Mécanisme de **fragmentation** et de réassemblage
- De même il faut savoir comment faire quand des processus insistent pour envoyer des unités de données trop petites pour être envoyées chacune séparément de façon efficace
  - Grouper plusieurs messages (**concaténation**) en un grand message

# Modèle: Principes

- La couche sous jacente peut décider d'utiliser la même connexion (**multiplexage**) pour plusieurs conversations indépendantes
  - Le multiplexage est de règle au niveau de la couche physique lorsque le trafic de toutes les connexions ne passent que sur un seul support
- Quand il existe plusieurs chemins possibles entre une source et une destination, il faut *choisir une route*
- Quelques fois cette décision est partagée entre deux couches ou plus
- Par exemple, pour envoyer des données de la Grande Bretagne vers l'Italie, une décision de *haut niveau* quant au passage via la France ou l'Allemagne selon les tarifs en vigueur et une décision de *bas niveau* doit préciser en fonction du trafic quel circuit utiliser

# Modèle: Terminologie

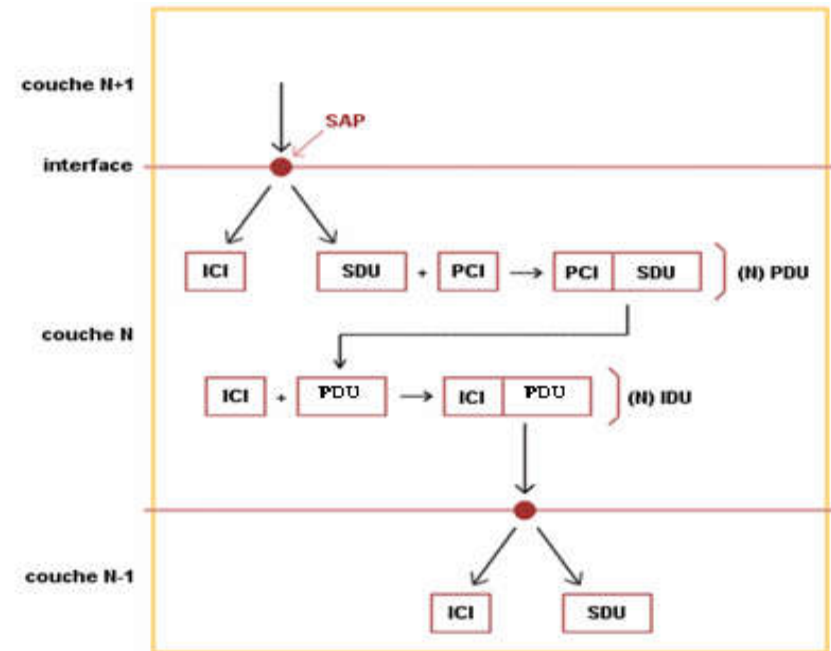
- Les éléments actifs de chaque couche s'appellent des **entités**
- Une entité peut être logicielle (processus) ou matérielle (Entrée/Sortie)
- Les entités de la même couche sur les machines différentes sont appelées **entités paires**
- Les entités de la couche  $n$  implémentent un service pour utilisé par la couche  $n+1$
- La couche  $n$  est alors appelée **fournisseur de service** et la couche  $n+1$  **utilisateur de service**
- La couche  $n$  peut utiliser les services de la couche  $n-1$  pour fournir ses services
- Elle peut offrir différentes catégories de service
- Les services sont accessibles par des **points d'accès aux services, SAP** (*Service Access Point*)

# Modèle: Terminologie

- Les SAP de la couche  $n$  sont les endroits où la couche  $n+1$  peut accéder aux services offerts
- Chaque SAP est identifié par une adresse unique
- Pour faire l'analogie avec le téléphone: SAP du système téléphonique sont les prises auxquelles peuvent être raccordées les combinés et les adresses SAP sont les numéros de téléphone de ces prises

# Modèle: Terminologie

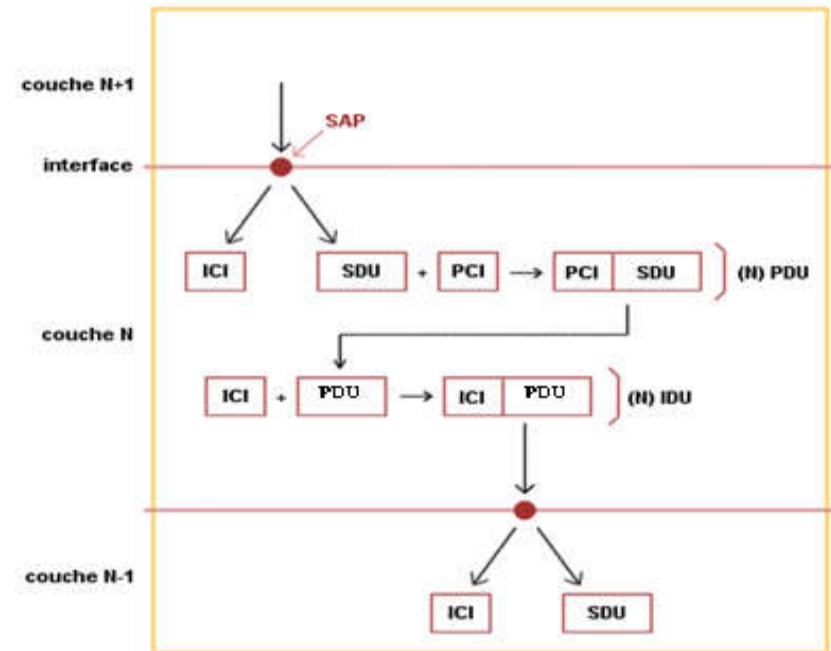
- Pour que deux couches échangent des informations, il doit exister un ensemble de règles admises en ce qui concerne l'interface
- Pour une interface, l'entité de la couche  $n+1$  passe une **unité de données d'interface IDU (*Interface Data Unit*)** à l'entité de la couche  $n$  par le SAP



- L'IDU est composé d'une **unité de données de service SDU (*Service Data Unit*)** et de certaines **informations de contrôle ICI (*Interface Control Information*)**
- Le SDU est l'information passée par le réseau à l'entité paire puis vers le haut à la couche  $n+1$
- L'information de contrôle est utilisée pour assister la couche inférieure dans son travail (le nombre d'octets par exemple), mais elle ne fait pas partie des données

# Modèle: Terminologie

- Pour transmettre une SDU, l'entité de la couche  $n$  peut devoir la fragmenter en différents morceaux
- Chacun reçoit un *en-tête* et est envoyé comme une **unité de données de protocole PDU (Protocol Data Unit)**



- Les en-têtes ou **information de contrôle du protocole PCI (Protocol Control Information)** sont utilisés par les entités paires pour transporter leur protocole pair
- Les PCI fournissent les numéros de séquence, les comptages etc.

# Modèle: Services

- Les couches peuvent offrir deux types différents de services aux couches supérieures: Le mode *connexion* et le mode *sans connexion*
- Dans le cas d'un service en mode connexion, l'utilisateur établit au préalable une connexion, l'utilise puis la relâche
- Le téléphone est un service en mode connexion
  - Pour parler à quelqu'un, on décroche le combiné, on compose le numéro puis on parle avant de raccrocher
- L'aspect essentiel de la connexion est qu'elle fonctionne comme un tuyau: l'émetteur pousse les bits d'un côté et le récepteur les récupère dans le même ordre de l'autre côté
- Un service sans connexion est structuré comme le système postal
  - Chaque message est acheminé indépendamment de tous les autres (l'ordre peut ne pas être respecté)
- Chaque service peut se caractériser par une **qualité de service**



# Modèle: Services

- Des services sont fiables quand ils ne perdent jamais de données
- La mise en œuvre d'un service fiable se fait par **acquiescement** (*accusé de réception*) de chaque message par le récepteur
- Le processus d'acquiescement engendre cependant une charge et un délai supplémentaire qui peuvent être parfois indésirables
- Les usagers du téléphone préfèrent entendre un peu de bruit ou un mot déformé que de devoir supporter un délai d'attente en raison des acquiescements
- De même pour de la vidéo, on peut accepter quelques pixels erronés alors qu'il serait insupportable que le film se déroule par saccades sous prétexte que l'on veut corriger les erreurs de transmission

# Modèle: Services

- Les services en mode connexion sont par **messages** ou par **flots**
- Lorsque le service est par message, les frontières des messages sont préservées
- Par exemple, quand on envoie deux messages de 1 Ko, ils arrivent sous la forme de deux message de 1 Ko et jamais comme un message de 2 Ko
- Dans le mode par flots, il n'y a pas de frontières de messages. Quand les 2 Ko arrivent chez le récepteur, il n'y a aucun moyen de dire s'ils proviennent d'un message de 2 Ko, de deux messages d'1 Ko et de 2048 messages d'1 octet
- Toutes les applications ne nécessitent pas le mode connexion
- Parfois, tout ce qu'il faut c'est pouvoir envoyer un message qui a une grande probabilité d'arrivée, mais sans garantie
- Ce type de service sans connexion non fiable (sans acquittement) est souvent appelé **service datagramme**

# Modèle: Primitives de services

- Un service est défini par un ensemble de **primitives** (opérations) qu'un utilisateur ou d'autres entités peuvent utiliser pour accéder au service
- Ces primitives ordonnent au service de réaliser telle action ou de rendre compte à la suite d'une mesure prise par l'entité paire
- Les primitives sont réparties suivant quatre classes:
  - **Request**: Une entité sollicite un service
  - **Indication**: Une entité est informée d'un événement
  - **Response**: Une entité répond à un événement
  - **Confirm**: Une entité a bien reçu la réponse à la demande
- Voyons comment on peut établir ou relâcher une connexion
  - L'entité initiatrice de la communication fait une demande de connexion (**CONNECT.request**), ce qui se traduit par l'envoi d'un paquet
  - Le destinataire reçoit alors une **CONNECT.indication** lui indiquant que quelqu'un veut établir une connexion avec lui
  - Le récepteur utilise la primitive **CONNECT.response** pour dire s'il accepte ou rejette la proposition de connexion

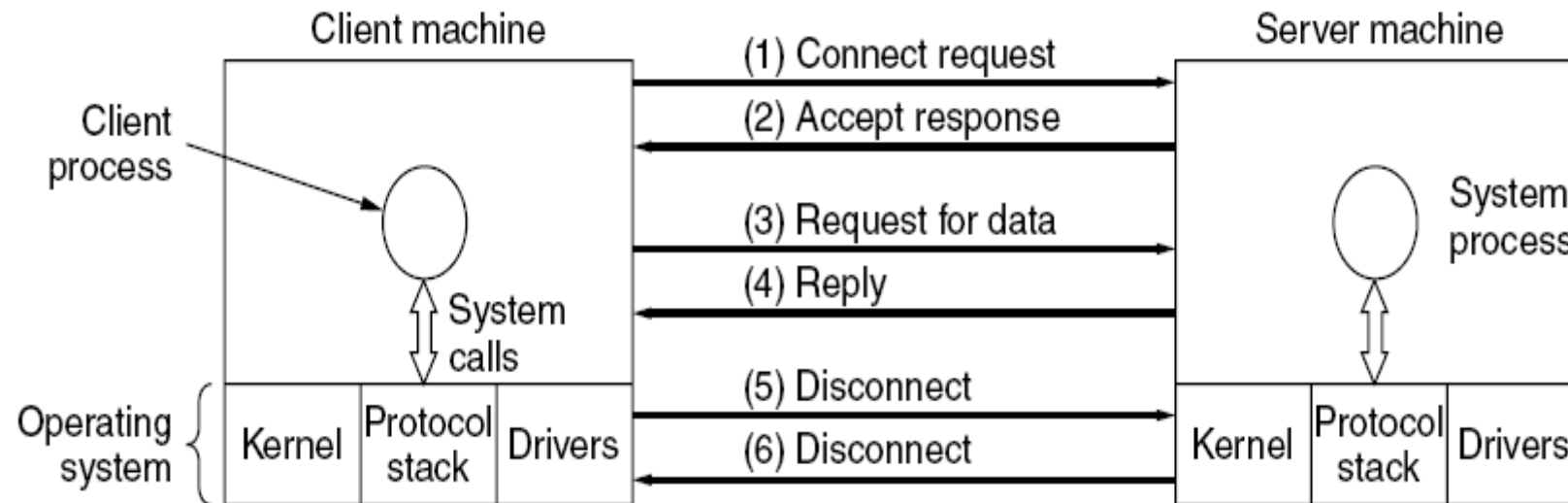
# Modèle: Primitives de services

- Voyons comment on peut établir ou relâcher une connexion
  - Dans tous les cas, l'entité émettrice est avertie de ce qui s'est passé via la primitive **CONNECT.confirm**
- La plupart des primitives possèdent des paramètres
- Les paramètres d'un **CONNECT.request** spécifient
  - La machine à laquelle on désire se connecter
  - Le type de service désiré
  - La taille maximale des messages utilisée avec cette connexion
- Si l'entité réceptrice n'est pas d'accord avec la taille des messages, elle peut faire une contre proposition dans sa primitive de réponse
- Le détail de cette **négociation** fait partie du protocole
- Par exemple, dans le cas de deux propositions conflictuelles sur la taille des messages, le protocole peut spécifier que c'est toujours la plus petite valeur qui est choisie

# Modèle: Primitives de services

- Les services sont **confirmés** ou **non confirmés**
- Service confirmé: il y a une demande, une indication, une réponse et une confirmation
- Service non confirmé: il y a juste une demande et une indication
- CONNECT est toujours un service confirmé parce que l'entité paire doit être d'accord pour établir la connexion
- Le transfert de données peut-être confirmé ou non, selon que l'émetteur a besoin ou non d'acquittement
- Exemple d'utilisation des primitives avec le réseau téléphonique
  - CONNECT.request: Composition du numéro
  - CONNECT.indication: Sonnerie chez le correspondant
  - CONNECT.response: Décrocher le combiné
  - CONNECT.confirm: Arrêt de la sonnerie chez l'appelant
  - DATA.request: Paroles
  - DATA.indication: Réception des paroles
  - DISCONNECT.request: Raccrocher le combiné
  - DISCONNECT.indication: Le récepteur raccroche également

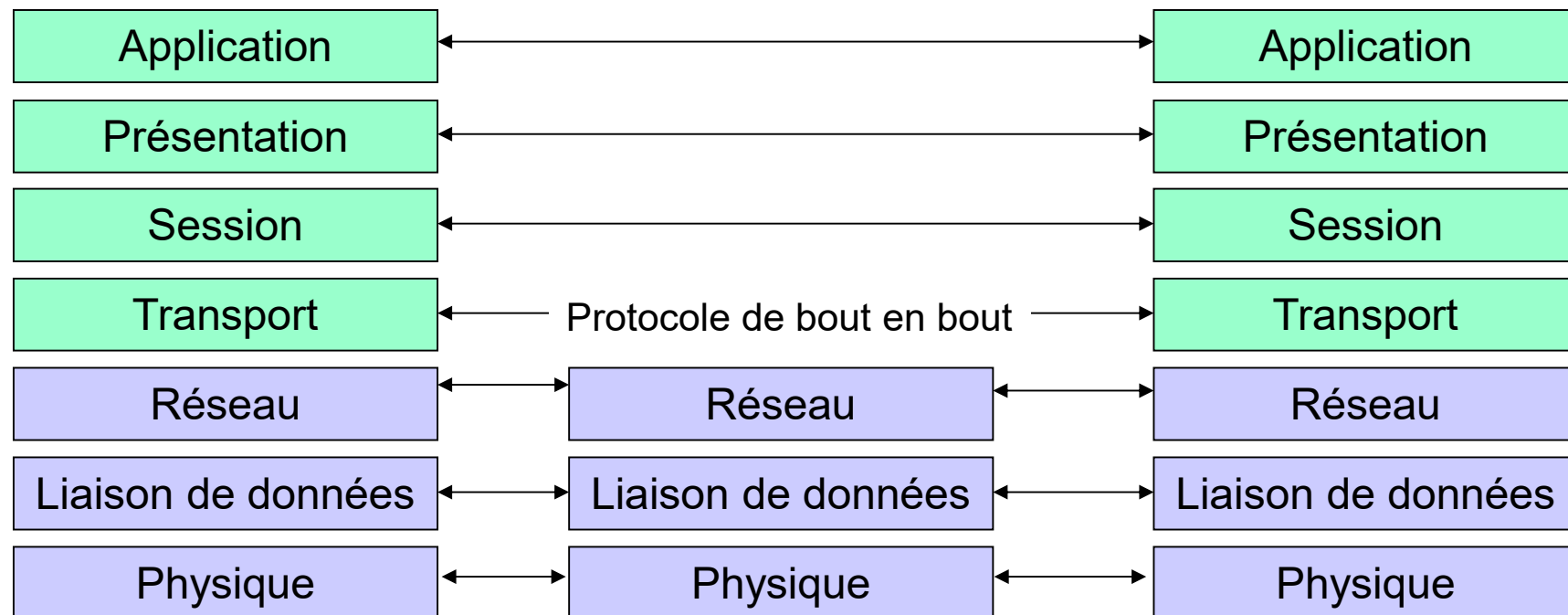
# Modèle: Primitives de services



# Modèle: Acronymes

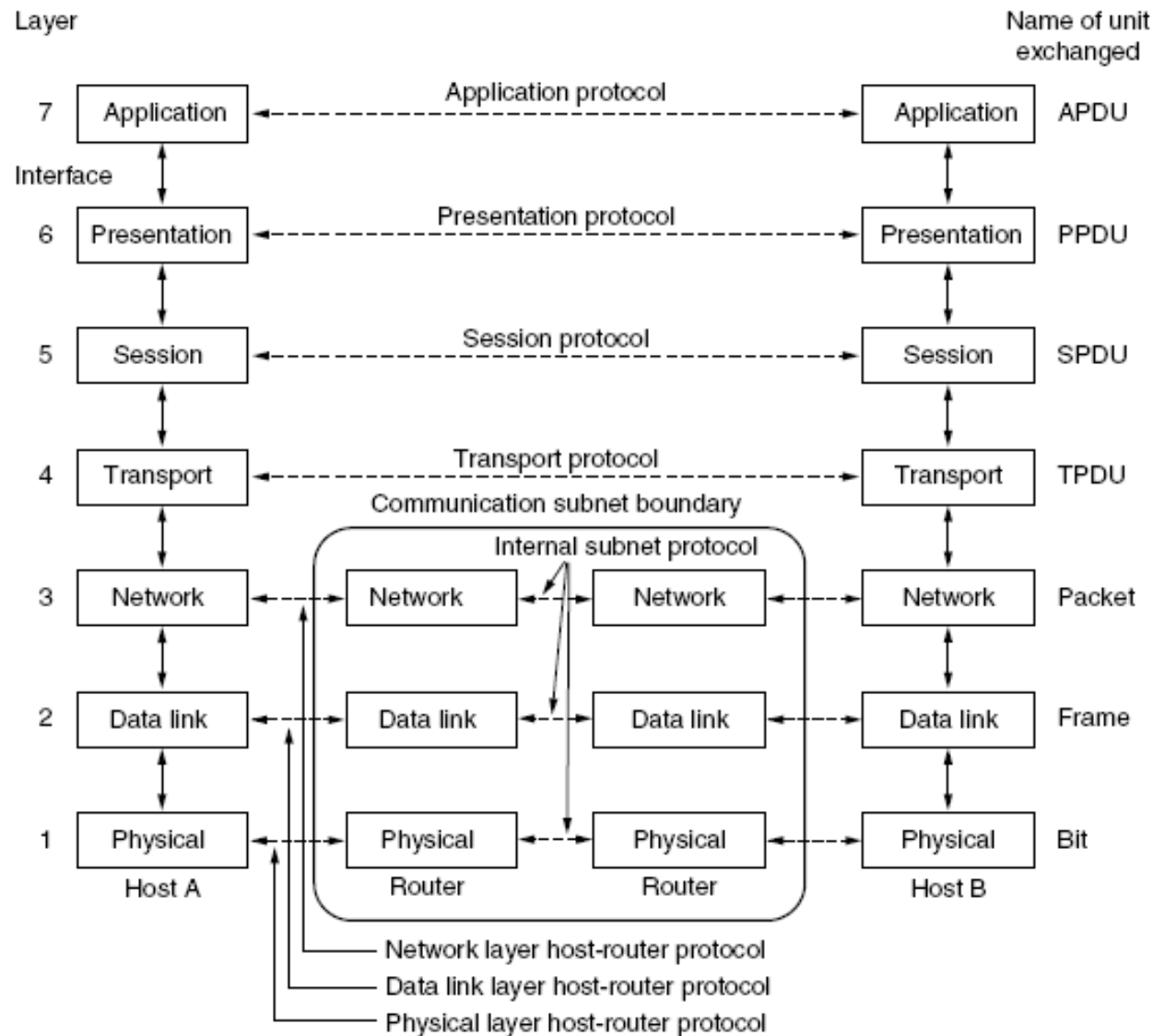
- **OSI**: Open System Interconnection
- **ISO**: International Organization for Standardization (AFNOR, ANSI,...)
- **IUT**: International Union of Telecommunication (ex CCITT: Comité Consultatif International Télégraphique et Téléphonique)
- **CCITT**: AVIS X, V, T
- **IETF**: Internet Engineering Task Force
- **IEEE**: Institute of Electrical and Electronics Engineers
- Modèle OSI : **ISO 7498-1**

# Modèle de référence: OSI





# Modèle de référence: OSI



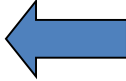
# Modèle de référence: OSI

- Le modèle OSI a 7 couches
- Les principes ayant conduit à leur élaboration sont:
  - Une couche doit être créée lorsqu'un nouveau niveau d'abstraction est nécessaire
  - Chaque couche exerce des fonctions bien définies
  - Les fonctions de chaque couche doivent être choisies en visant la définition de protocoles normalisés
  - Le choix des frontières entre couches doit minimiser le flux d'information aux interfaces
  - Le nombre de couches doit être suffisamment grand pour éviter la cohabitation dans une même couche de fonction très différentes et suffisamment petit pour éviter que l'architecture ne devienne difficile à maîtriser

# Modèle de référence: OSI

- La couche (N) est composée de:
  - Le service (N): événements à mettre en place pour rendre un service au niveau supérieur
  - Le protocole (N): règles et conventions utilisées pour gérer la conversation entre deux entités
  - Les points d'accès au service (N) ou (N)-SAP
- Service et protocole:
  - Sémantique d'association
  - Sémantique de fonctionnalité
  - La syntaxe

# Modèle de référence: OSI

- Sémantique d'association
    - Mode avec connexion
    - Mode sans connexion
  - Primitives de service
    - (DIS).CONNECT.request
    - (DIS).CONNECT.indication
    - (DIS).CONNECT.response
    - (DIS).CONNECT.confirm
  - Sémantique de fonctionnalité
    - Fragmentation- réassemblage, concaténation- séparation
    - Contrôle d'erreur, remise en séquence, contrôle de flux,...
- 
- Commutation de circuits
  - Commutation de messages
  - Commutation de paquets
- DATA.request
  - DATA.indication

# Modèle de référence: OSI

- Transmission des bits sur un circuit de communication
- Éléments de la couche physique
- Support, codeurs et démodulateurs, multiplexeurs et concentrateurs
- PHY (Physical)
  - Correspondance entre signal reçu et son interprétation sous forme binaire
- PMD (Physical Medium Dependant)
  - Description du média utilisé, câbles, prises, ...



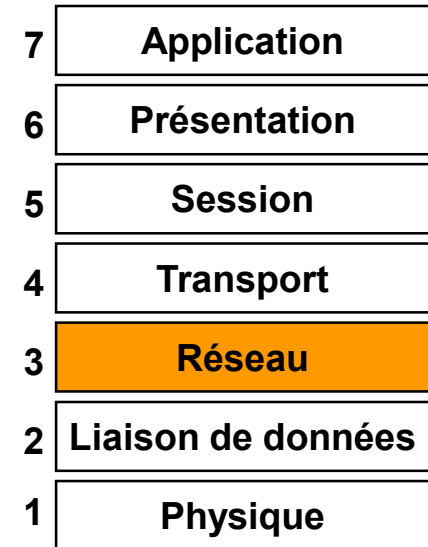
# Modèle de référence: OSI

- Liaison de données
  - Utilise la couche physique
  - Gestion de la liaison de données
    - Données de l'émetteur en trame de données
    - Transmission des trames en séquence
    - Gestion des trames d'acquittement
    - Reconnaissance des frontières de trames envoyées par la couche physique
  - Détection et reprise sur erreur
    - Régulation du trafic
    - Gestion des erreurs
  - MAC (Medium Access Control)
  - LLC (Layer Link Control)



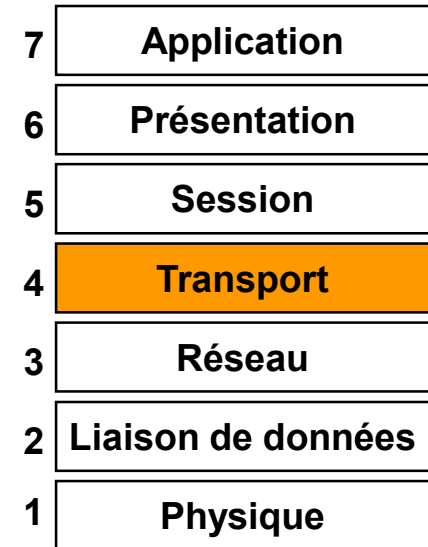
# Modèle de référence: OSI

- Réseau
  - Fournit les moyens d'établir, de maintenir, et de libérer des connexions de réseau entre systèmes ouverts
    - Gestion de sous réseau
    - Acheminement des paquets (T-PDU) source vers la destination
  - Fonctionnalités
    - Adressage
    - Routage
  - Mode connecté / non connecté
    - X25, IP



# Modèle de référence: OSI

- Transport
  - Indépendance des réseaux sous-jacents
  - Accepte les données de la couche session
    - S'assure de l'ordonnancement
    - Segmentation
  - Contrôle de flux
  - Contrôle/récupération d'erreurs
  - Fonctionnalités de bout en bout
    - Multiplexage de plusieurs messages sur un canal
  - Protocoles de transport
    - TCP, UDP





# Modèle de référence: OSI

- Session
  - Responsable de la synchronisation
  - Fonctions de type
    - Points de reprise
    - Retour arrière



# Modèle de référence: OSI

- Présentation
  - Représentation des données transférées entre applications
  - Représentation de la structure de données et représentation de l'ensemble des actions effectuées sur cette structure de données
  - Encodage dans une norme agréée permettant aux équipements de communiquer
    - Compression, chiffrement
    - Exemple: la syntaxe abstraite ASN.1 normalisée par l'ISO
      - Messageries X400, annuaires X500



# Modèle de référence: OSI

- Application
  - Offre aux processus d'application le moyen d'accéder à l'environnement OSI
  - Les processus d'applications échangent leurs informations par l'intermédiaire des entités d'applications
    - Ex: terminal de réseau virtuel, transfert de fichiers, courrier électronique, consultation annuaire



# Modèle TCP/IP

