

CM 1:

Introduction aux

Réseaux de Données

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

Ramon APARICIO-PARDO

Ramon.Aparicio-Pardo@unice.fr

25/09/2020

Organisation de la matière Systèmes et Réseaux (S5)

❖ Deux parties:

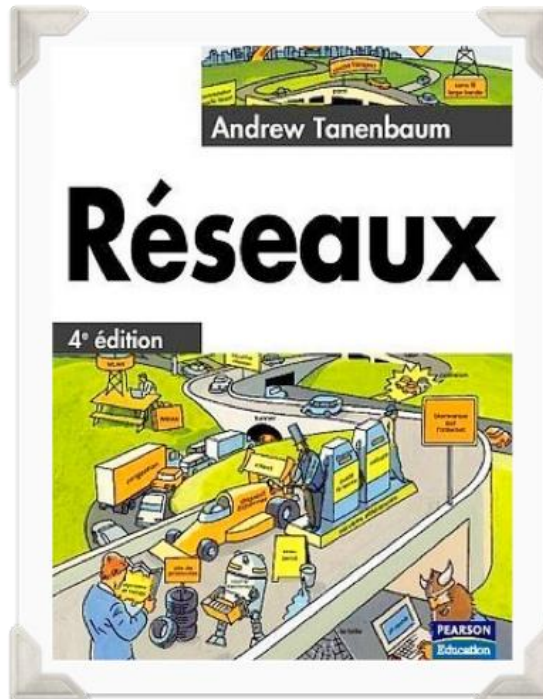
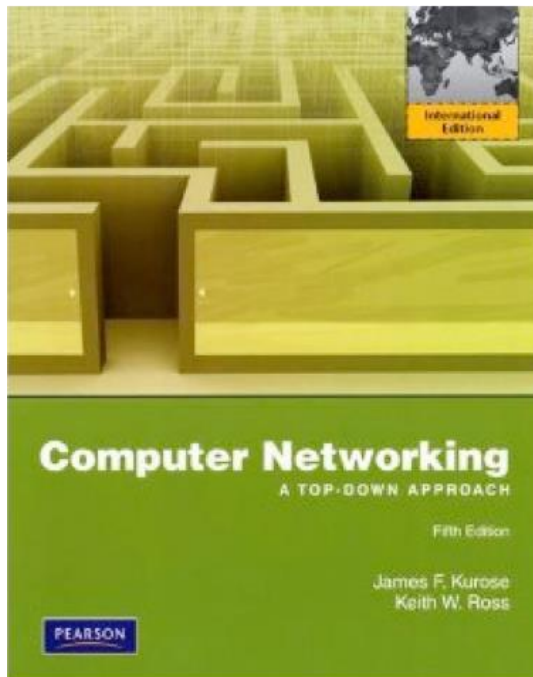
1. Systèmes (Gilles MENEZ)

2. Réseaux (Ramon APARICIO)

- Remplacé cette année par Michel SYSKA
- Supports:

<https://github.com/uns-iut-info/doc-miage-network-2020>

Bibliographie

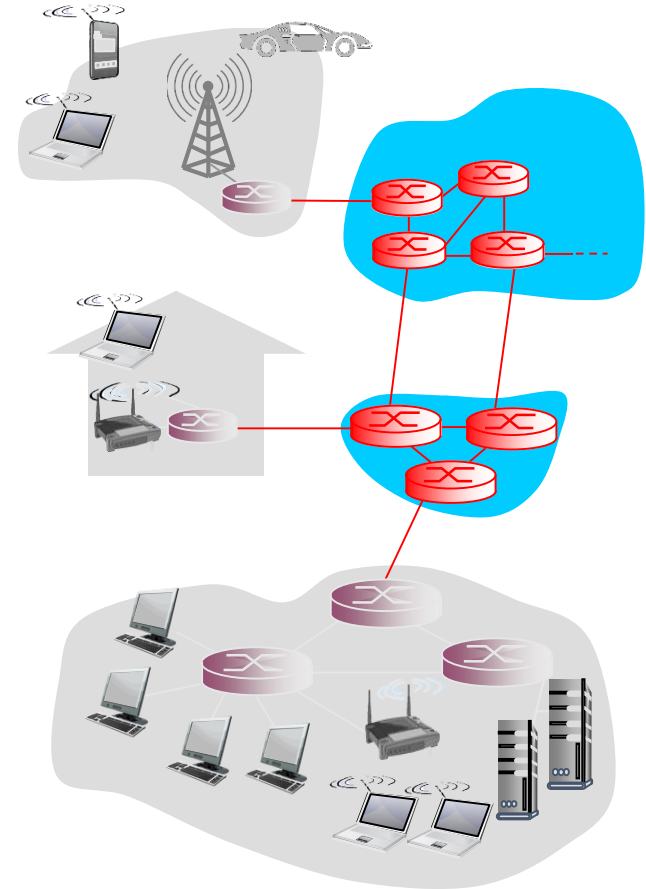


PLAN CM 1

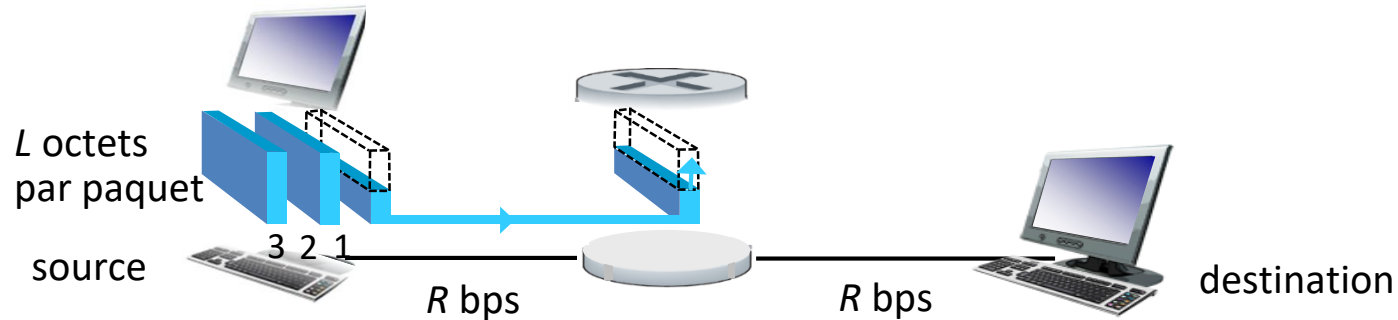
- 1. COMMUTATION DE PAQUETS**
- 2. PERFORMANCES DES RÉSEAUX DE DONNÉES**
- 3. FILES D'ATTENTES**
- 4. PROTOCOLES DES RÉSEAUX**

Réseaux de données : réseaux à commutation de paquets

- ❖ Réseaux de données formés par des *dispositifs de commutation de paquets* (les nœuds) interconnectés par des liens
- ❖ **Commutation de paquets:** les hôtes découpent les messages de données de la couche applicative en morceaux plus petits (paquets) :
 - Les paquets sont réexpédiés d'un dispositif à l'autre à travers les liens sur le chemin qui lie la source du paquet à sa destination
 - Chaque paquet est transmis à pleine capacité du lien

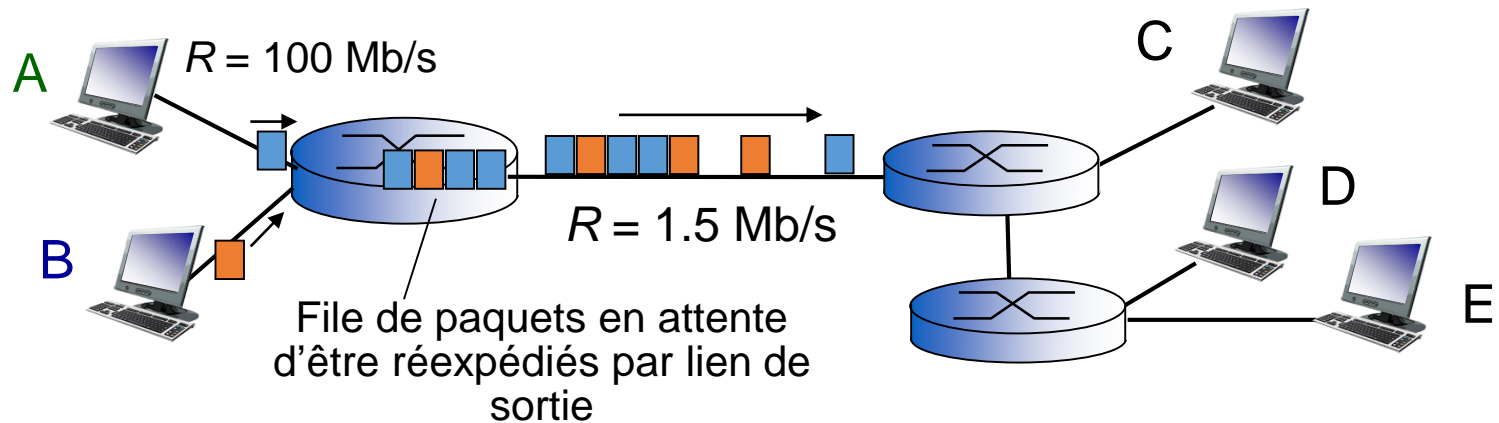


Commutation de paquets : *store-and-forward*



- ❖ Il faut L / R secondes pour transmettre un paquet de taille L bits sur un lien avec une capacité de R bps
- ❖ ***Store-and-forward*** : Le paquet entier doit arriver au dispositif de commutation avant qu'il puisse être transmis sur le lien suivant
- ❖ Délai de bout en bout: nombre liens $\times (L / R)$
 - (en supposant zéro délai de propagation)
 - P.ex. $L = 7.5 \text{ Mb}$, $R = 1.5 \text{ Mbps}$, 2 sauts
 - $\text{délai} = 2 \times (7.5/1.5) = 2 \times 5 \text{ s} = 10 \text{ s}$

Commutation de paquets : *délai d'attente, perte*



❖ Files d'attente et pertes:

- Si le taux d'arrivée (en bits/s) au lien de sortie de dispositif dépasse le taux de transmission du lien pendant une période de temps :
 - Les paquets seront mis en attente dans la file, avant d'être transmis sur le lien
 - Les paquets peuvent être supprimés (perdus) si la file (tampon) se remplit

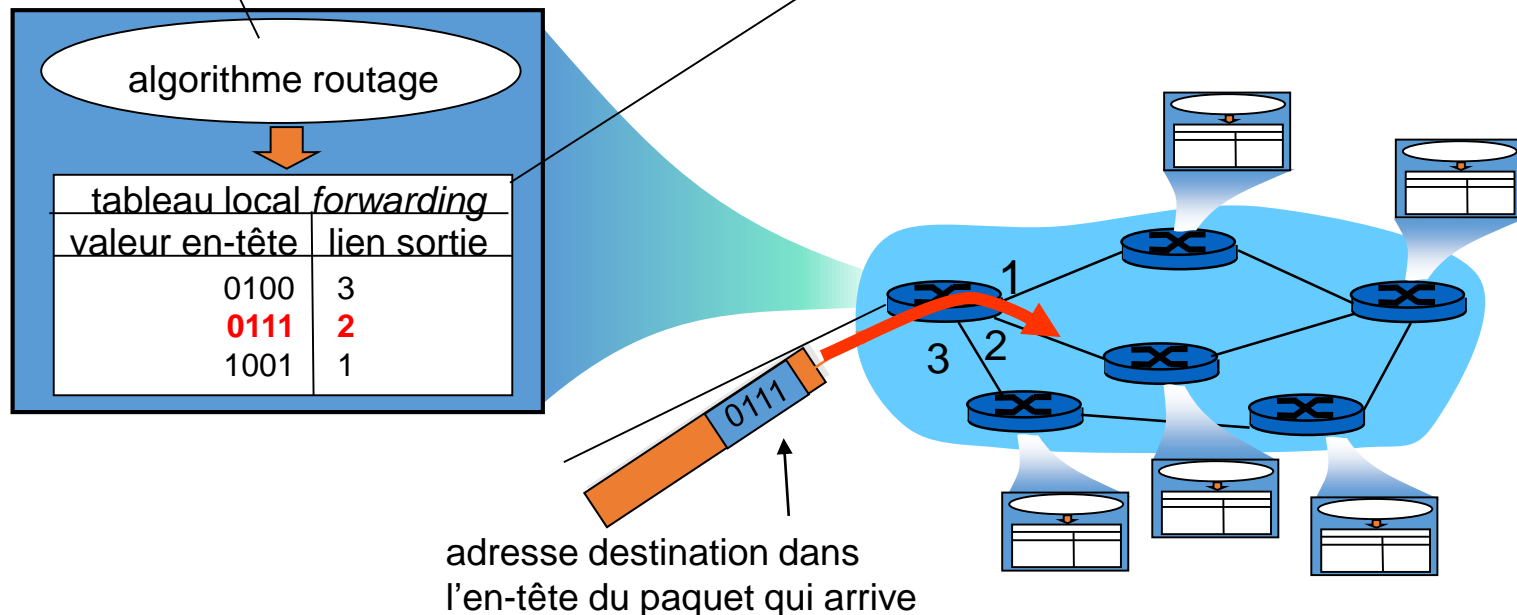
Commutation de paquets : *routing et réexpédition*

❖ **Routing (routing):**

- Il détermine la route source-destination prise par les paquets :
- *Algorithmes de routage (CM3)*

❖ **Réexpédition (forwarding):**

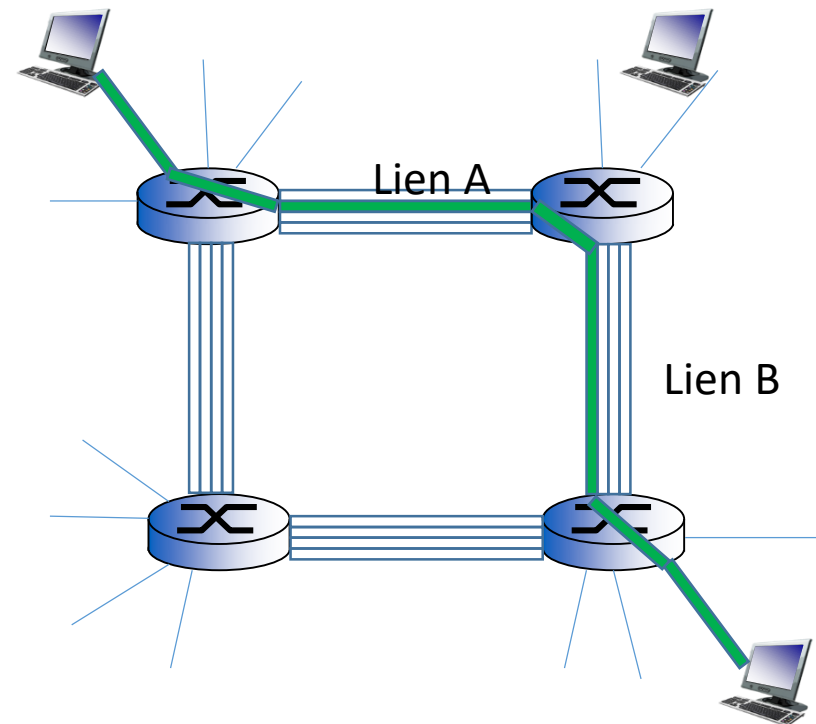
- Il déplace (*commute*) les paquets de une entrée du dispositif vers la sortie appropriée du même



Commutation de paquets vs Commutation de circuits

❖ Commutation de circuits :

- Ressources affectées de bout en bout:
- Réservées pour «l'appel» entre source et destination
- ❖ Dans le diagramme, chaque lien a quatre circuits
 - Appel obtient 2^e circuit sur le lien A et 1^{er} circuit sur le lien B.
- ❖ Ressources dédiées: ***pas de partage***
 - Performances garanties
- ❖ Circuit inactif, si non utilisé par appel (*pas de partage*)
- ❖ Paradigme généralement utilisé dans *les réseaux téléphoniques classiques*



Commutation de paquets vs Commutation de circuits

❖ *La commutation de paquets, est-elle un gagnante à coup sûr*

✓ **Idéal pour les données en rafale (salves des données)**

- Partage de ressources
- Plus simple, pas d'établissement d'appel

X **Congestion excessive possible:** retard de paquet et perte

- Protocoles nécessaires pour un transfert fiable des données, contrôle de congestion

❖ **Q: Comment fournir un comportement de type circuit?**

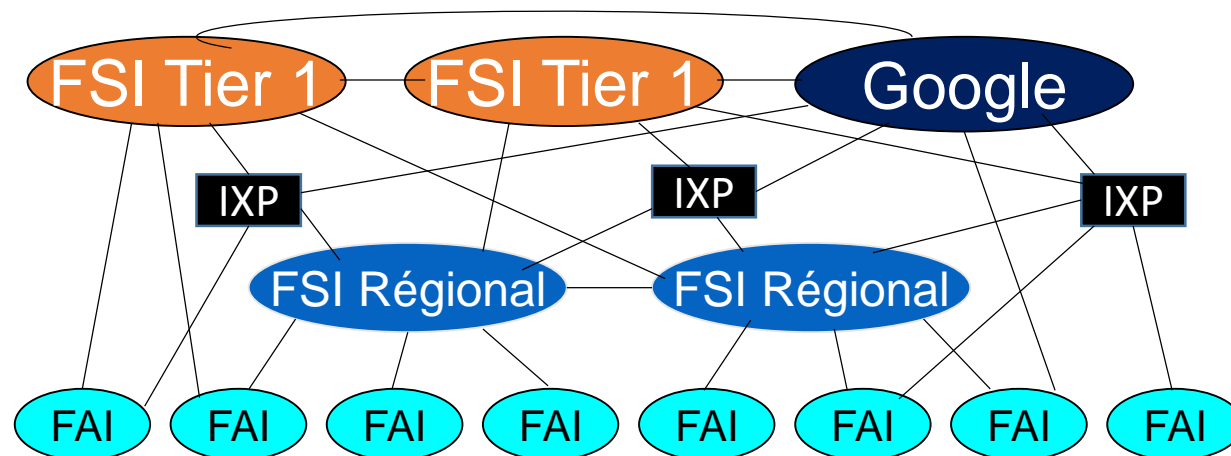
- Garantie de bande passante nécessaire pour les applications audio / vidéo
- Toujours un problème non résolu

❖ **Analogie :**

- Ressources réservées (*commutation de circuit*) par rapport à l'allocation à la demande (*commutation par paquets*)

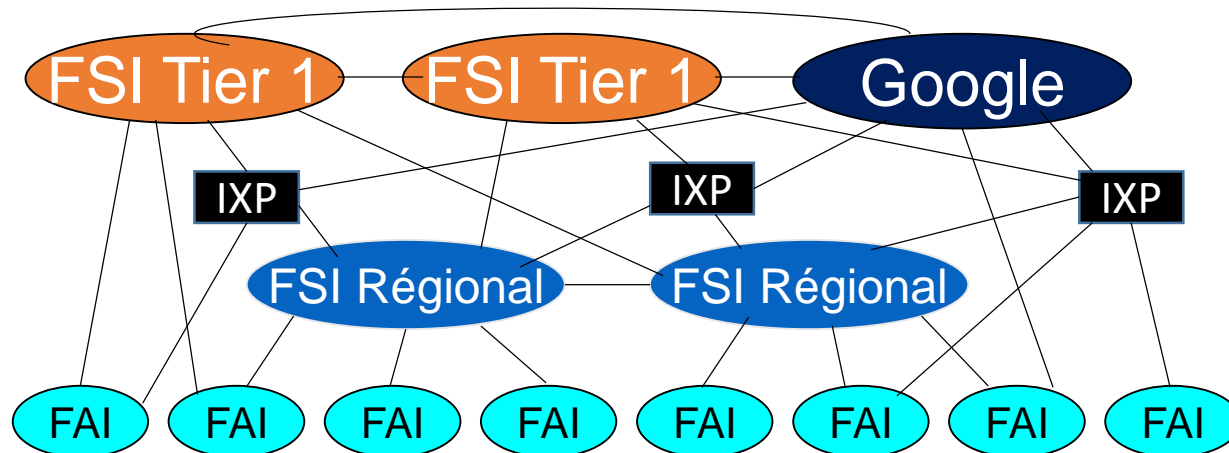
Internet : réseau des réseaux

- ❖ Systèmes d'extrémité se connectent à Internet via FAI (Fournisseurs d'Accès Internet)
 - Les FAI peuvent être privés (ex. Orange), d'entreprise et universitaires (ex. UCA)
- ❖ Les FAI doivent à leur tour être interconnectés.
 - Pour que deux hôtes puissent envoyer des paquets les uns vers les autres
 - Par le moyen de une hiérarchie d'autres types des réseaux
- ❖ Ce réseau des réseaux résultant est très complexe
 - Evolution a été guidée par l'économie et les politiques nationales



Internet : réseau des réseaux

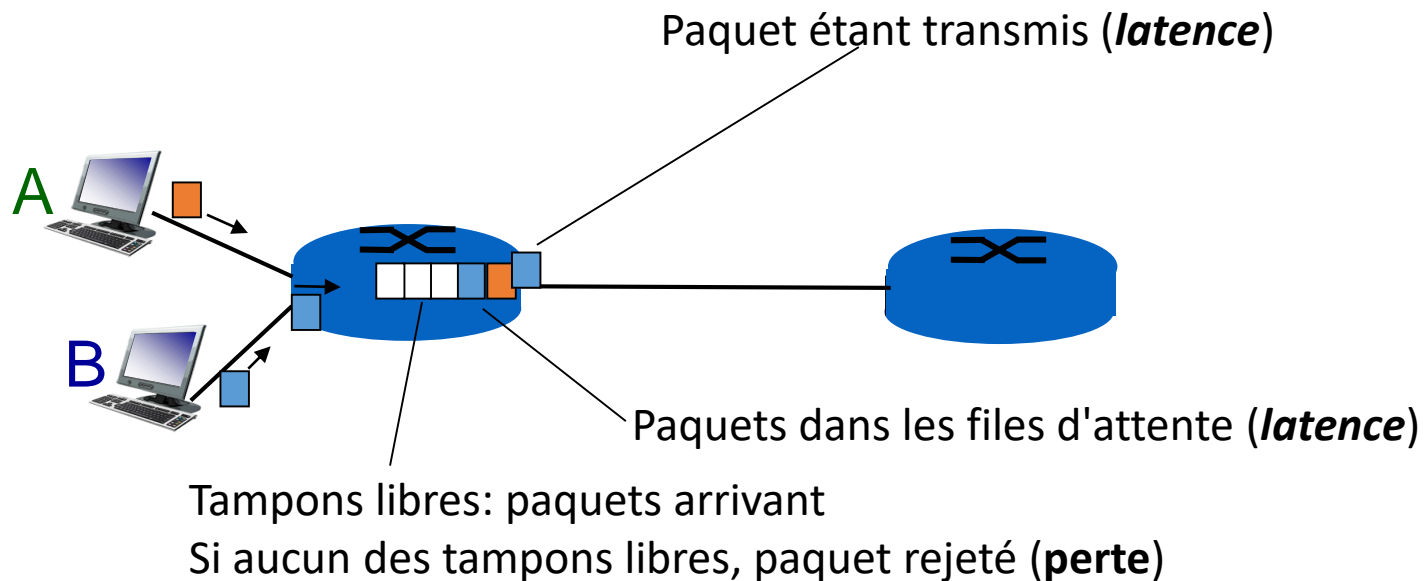
- ❖ Au centre: petit nombre de grands réseaux bien connectés
 - FSIs (Fournisseur de Service Internet) de niveau 1 (*Tier 1 ISP*) commerciaux (ex.: Niveau 3, Sprint, AT&T, Orange),
 - couverture nationale et internationale
 - Réseaux des fournisseurs des contenus (*Content Providers, CP*) (ex.: Akamai, YouTube, Google):
 - réseaux privés qui relient leurs centres de données à Internet, en contournant souvent tier-1, les FAI régionaux.



Performances des réseaux de données : latences, pertes, débits

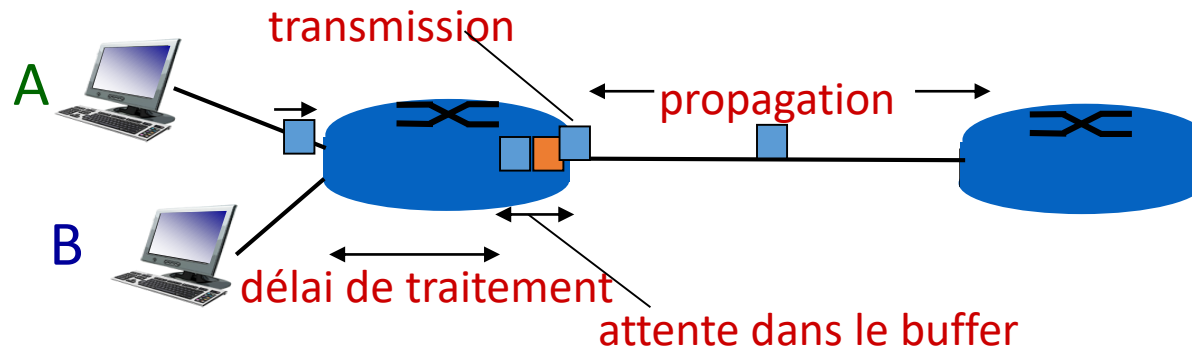
❖ Comment se produisent les **pertes** et les **latences**?

- Les paquets attendent dans les tampons (**buffers**) du dispositif de commutation
- Le débit (ou taux) d'arrivée de paquets (temporairement) dépasse la capacité du lien de sortie
- Les paquets attendent leur tour pour accéder au lien de sortie



Latences

- ❖ **Latence**: temps nécessaire à un paquet de données pour passer de la source à la destination à travers un réseau, c.-à-d., temps écoulé entre émission du 1^{er} bit du paquet à la source et la réception du dernier bit à la destination
 - *delay, latency*
 - Eventuellement le temps d'aller-retour (*RTT: round-trip time*)
- ❖ Composantes de la latence d'un paquet: (1) *durée transmission*, (2) *attente dans le buffer*, (3) *propagation* et (4) *délai de traitement dans le nœud*



$$latence = d_{\text{trait}} + d_{\text{buffer}} + d_{\text{trans}} + d_{\text{prop}}$$

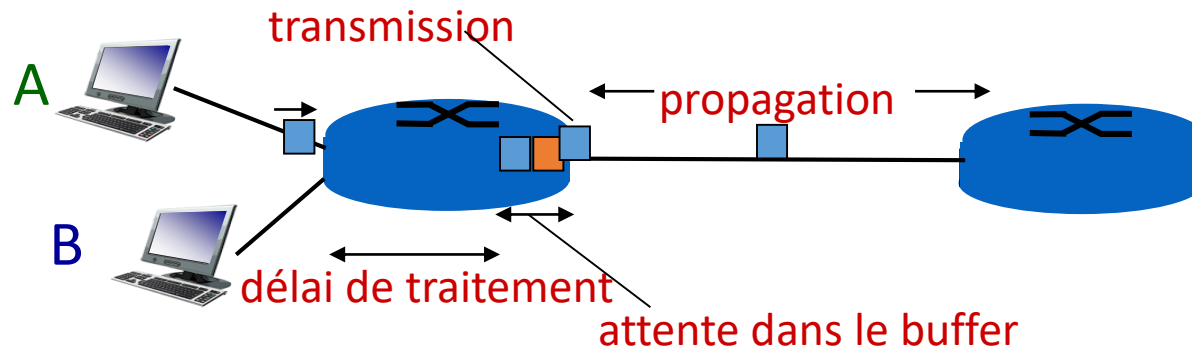
Latences

❖ Délai de traitement dans le nœud (d_{trait}):

- Vérifier les erreurs binaires (*checksum*)
- Traitement (lecture) des en-têtes
- Déterminer le lien de sortie (*forwarding*)
- *Magnitude: typiquement $< \text{msec}$*

❖ Temps d'attente dans le buffer (d_{buffer})

- Temps d'attente dans le tampon sur le lien de sortie pour la transmission
- *Magnitude: Dépendant du niveau de **congestion** du lien de sortie (cf. théorie de files d'attente)*



$$\text{latence} = d_{\text{trait}} + d_{\text{buffer}} + d_{\text{trans}} + d_{\text{prop}}$$

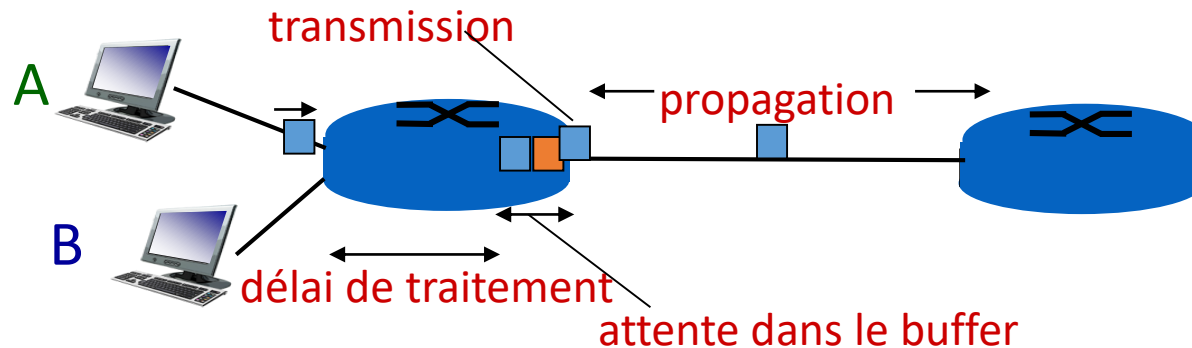
Latences

❖ Durée de transmission (d_{trans}):

- s : longueur de paquet (*bits*)
- r : bande passante de liaison (*bps*)
- *Magnitude*: $d_{trans} = s / r$

❖ Retard de propagation (d_{prop})

- l : longueur (distance) du lien physique (*m*)
- v : vitesse de propagation dans le milieu ($\sim 2 \times 10^8$ *m/s*)
- *Magnitude*: $d_{prop} = l / v$



$$latence = d_{trait} + d_{buffer} + d_{trans} + d_{prop}$$

Latences

- ❖ Parmi les quatre composantes, seule la ***durée de transmission*** dépend de la taille du paquet (du montant de bits à transmettre).
- ❖ Les trois autres constituent la ***latence de base*** c.-à-d., le temps écoulé avant de recevoir le 1er bit du paquet.
- ❖ *Exemple: Message bref (ping) transocéanique*
 - Envoi de 56 octets (ping) à une distance de 10 000 km
 - Délai des traitements et d'attentes négligeables ($d_{\text{trait}}, d_{\text{buffer}} \sim 0$)
 - Débit de l'accès Internet : 10 Mbps (ADSL) / 100 Mbps (Fibre)

Propagation : $d_{\text{prop}} = l / v = 10^7 \text{ (m)} / 2 \times 10^8 \text{ (m/s)} = 0,05 \text{ s} = \mathbf{50 \text{ ms}}$

Transmission (ADSL) : $d_{\text{trans}} = s / r = 56 \times 8 \text{ (bits)} / 10 \times 10^6 \text{ (bps)} \sim 5 \times 10^{-5} \text{ s} = \mathbf{0,05 \text{ ms}}$

Transmission (Fibre) : $d_{\text{trans}} = s / r = 56 \times 8 \text{ (bits)} / 100 \times 10^6 \text{ (bps)} \sim 5 \times 10^{-6} \text{ s} = \mathbf{0,005 \text{ ms}}$

Latence de base (propagation) domine, débit accès Internet à peu influence

Latences

- ❖ Parmi les quatre composantes, seule la ***durée de transmission*** dépend de la taille du paquet (du montant de bits à transmettre).
- ❖ Les trois autres constituent la ***latence de base*** c.-à-d., le temps écoulé avant de recevoir le 1er bit du paquet.
- ❖ *Exemple: Vidéo HD de 5 min (YouTube) transocéanique*
 - Envoi de 300 Mo à une distance de 10 000 km
 - Délai des traitements et d'attentes négligeables ($d_{\text{trait}}, d_{\text{buffer}} \sim 0$)
 - Débit de l'accès Internet : 10 Mbps (ADSL) / 100 Mbps (Fibre)

Propagation : $d_{\text{prop}} = l / v = 10^7 \text{ (m)} / 2 \times 10^8 \text{ (m/s)} = 0,05 \text{ s} = \mathbf{50 \text{ ms}}$

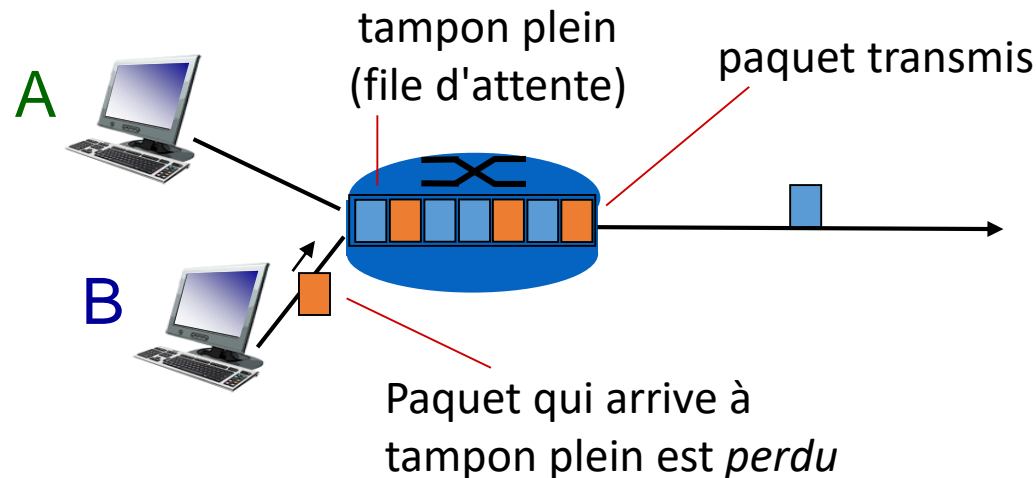
Transmission (ADSL) : $d_{\text{trans}} = s / r = 300 \times 8 \times 10^6 \text{ (bits)} / 10 \times 10^6 \text{ (bps)} = \mathbf{240 \text{ s}}$

Transmission (Fibre) : $d_{\text{trans}} = s / r = 300 \times 8 \times 10^6 \text{ (bits)} / 100 \times 10^6 \text{ (bps)} = \mathbf{24 \text{ s}}$

Débit domine, latence de base a peu d'influence

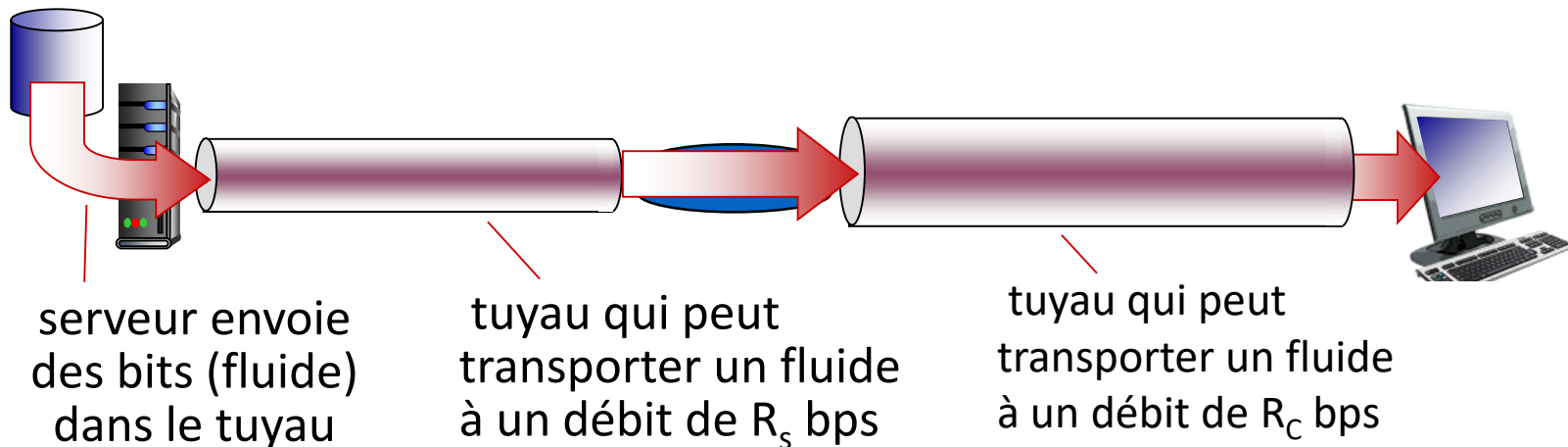
Perte de paquets

- ❖ La file d'attente (ou tampon) du lien de sortie a une taille finie
- ❖ Un paquet qui arrive sur une file d'attente pleine est rejeté (c.-à-d. pas admis, perdu)
- ❖ Le paquet perdu peut être retransmis par le nœud précédent, par la source ou pas du tout.



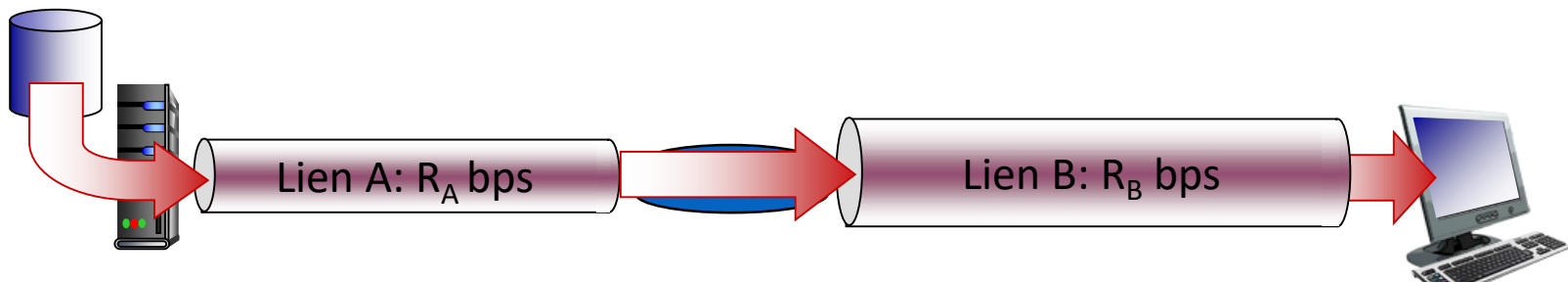
Débit

- ❖ **Débit (ou bande passante):** taux (bits par unité de temps) auquel les bits sont transférés entre l'émetteur et le récepteur
 - *bandwidth, throughput, bit rate*
 - *instantané:* taux au point donné dans le temps
 - *moyen:* taux sur une plus longue période de temps
- ❖ *Analogie avec le débit d'un tuyau*



Débit

- ❖ Quel est le débit moyen de bout en bout ?
 - Le débit du lien qui fait de goulot d'étranglement (*bottleneck link*)
- ❖ *Bottleneck link* : lien avec la capacité la plus petite
 - Dans l'exemple : le lien A ($R_A < R_B$)
 - Dans la pratique, le *bottleneck link* se trouve dans le dernier saut (lien d'accès à Internet)

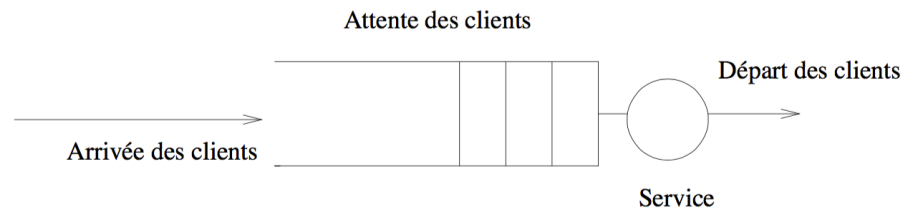


Simulation à Événements Discrets

- ❖ Si on veut étudier de manière plus précise les **pertes** et les **temps d'attente**, on doit faire recours à la modélisation des buffers des nœuds des commutation et à la simulation de leurs fonctionnements : la *théorie des files d'attente* et la *simulation à événements discrets*, respectivement.
- ❖ **Simulation à Événements Discrets** : c'est un type de simulation qui fait progresser l'horloge par étapes discrètes, souvent de taille irrégulière, plutôt que par de très petits pas réguliers. Ces étapes discrètes correspondent à l'intervalle de temps entre deux événements consécutifs.
- ❖ La simulation est *dirigée* par des *événements* :
 - À chaque étape, l'horloge avance vers l'événement suivant planifié dans **une file d'attente d'événements**, et l'événement est traité.
 - Étant donné que seuls les événements peuvent entraîner un changement de l'état de la simulation, il n'y a aucun intérêt à faire avancer l'horloge par des pas de temps plus petits que les intervalles entre les événements.
- ❖ Les simulations et modèles à événements discrets sont utilisés de façon classique pour modéliser des problèmes d'accès concurrents à une ressource partagée, c.-à-d. les problèmes des **files d'attente** :
 - *En dehors de l'informatique* : guichets de la poste, file d'attente du Resto'U, péage autoroutier, etc...
 - *Dans l'informatique* : bande passante, serveur, disque, mémoire, etc..

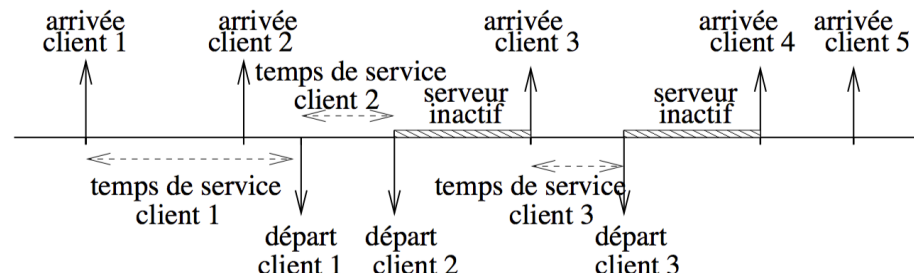
Modèles des files d'attentes

- ❖ **Modèles des files d'attentes:** Ils modélisent l'attente des clients pour accéder à une ressource (ou service) partagée (bande passante, serveur, disque, mémoire, ...) comme le temps écoulé entre l'arrivée à la file d'attente et le départ du système ($d_{\text{buffer}} + d_{\text{trans}}$), typiquement après avoir réussi à accéder au service.



- ❖ Donc, les différents **événements** sont:

- **Arrivée** du client au système et entrée dans la file d'attente (suivi, si possible, par le début du service)
- **Départ** du client du système, après la fin du service (suivi, si possible, par le début du service pour un nouveau client)



Modèles des files d'attente

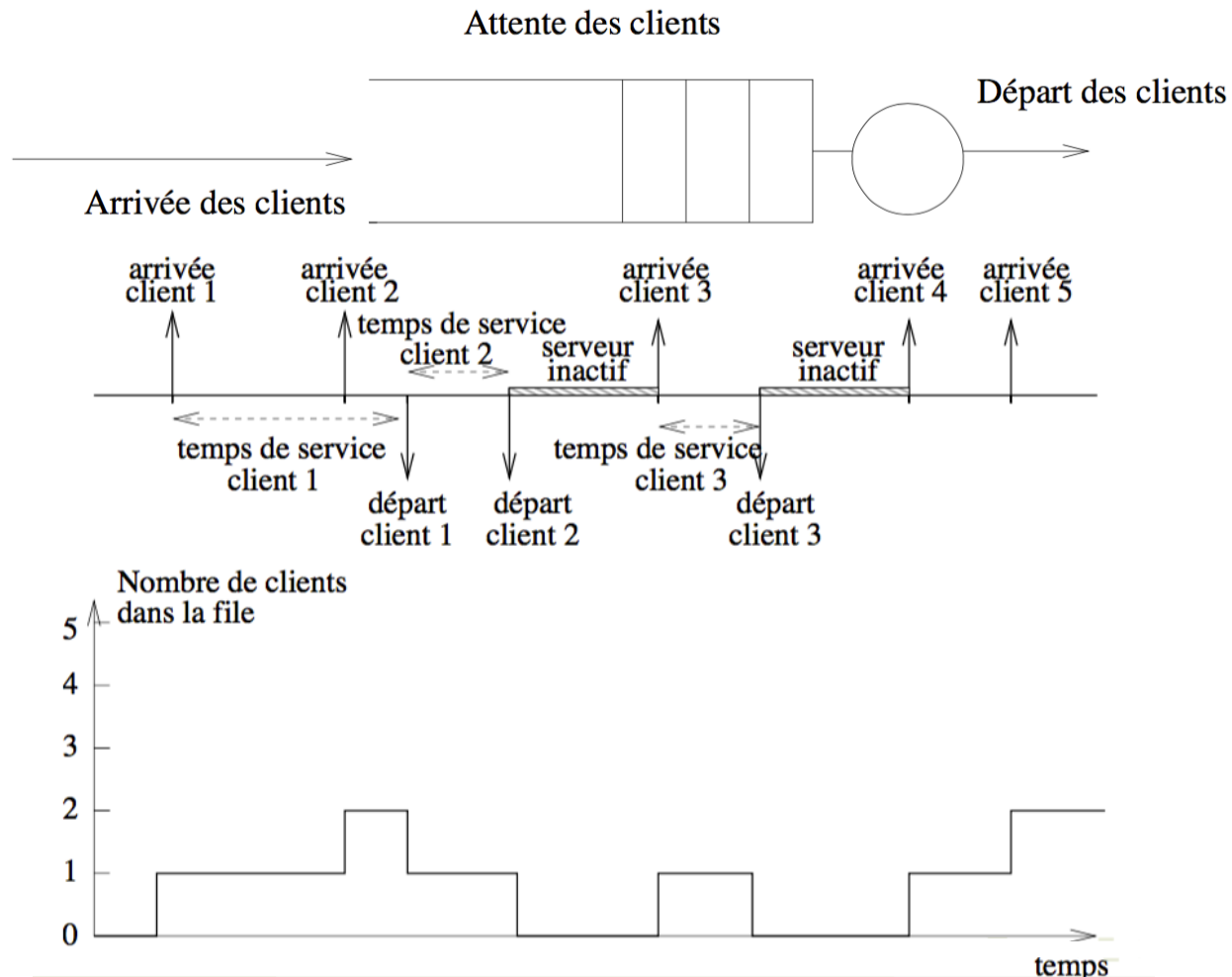
❖ Caractérisation du système :

- nombre maximum de clients dans le système (attente ou service)
- nombre de serveurs (mono-serveur / multiserveurs)
- arrivées des clients (ex : processus de Poisson)
- durée de services (ex : loi exponentielle)
- discipline de service (ex : FIFO=First In First Out)

❖ Modélisation mathématique :

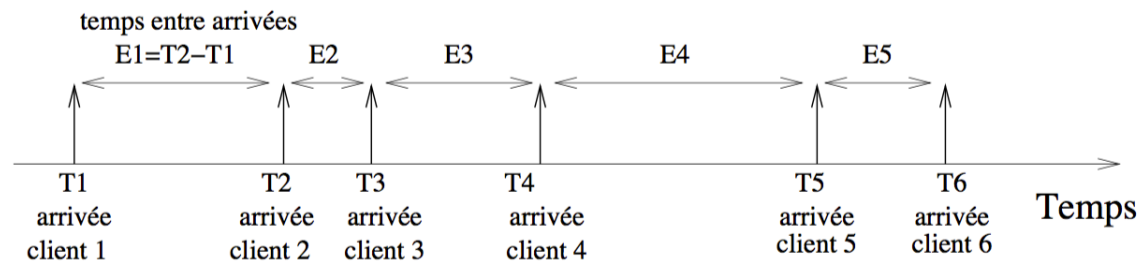
- **Elle est dehors des objectifs de ce cours, mais quelques mots ...**
- l'évolution au cours du temps du **nombre de clients dans le système (en attente ou en cours de service)**...
- ... est modélisée par un processus aléatoire (le plus souvent une chaîne de Markov) ...
- dont on étudie le régime stationnaire ...
- ... pour en déduire les performances moyennes (taux de blocage moyen, délai moyen, taux moyen d'utilisation des ressources, etc...)

Modèles des files d'attentes



Arrivées de clients

❖ Processus des arrivées des clients :



❖ Cas général et simple :

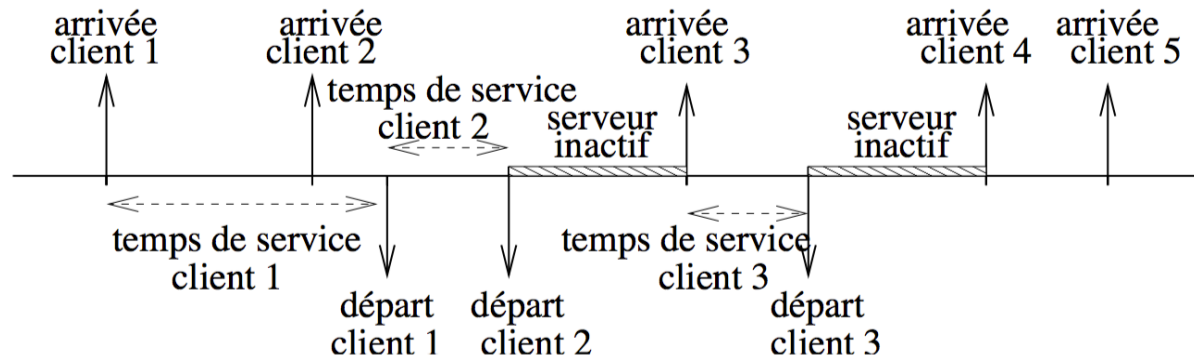
- Les temps d'inter-arrivées successifs E_1, E_2, E_3 , etc... *sont indépendants entre eux* et *ont même loi de probabilité*.
- Le processus des arrivées de clients est alors entièrement caractérisé par la loi des inter-arrivées.

❖ Cas particulier important : arrivées poissonniennes

- les inter-arrivées successives E_1, E_2, E_3 , etc... *sont indépendantes entre elles* et *distribuées selon une loi exponentielle de paramètre λ*
- de façon équivalente le processus des arrivées est un **processus de Poisson de paramètre λ**
- λ représente le nombre moyen d'arrivées par unité de temps (unité : clients/seconde)
- la durée moyenne des inter-arrivées vaut à $t_a = 1/\lambda$ (unité : seconde)

Durées de service

❖ Processus des départs des clients :



❖ Cas général et simple :

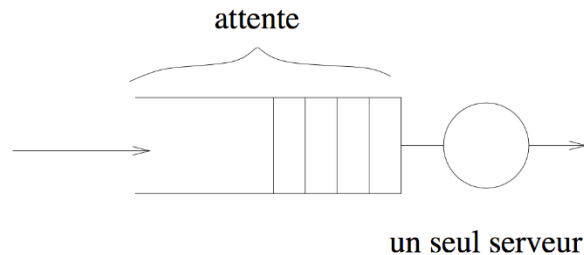
- En règle générale les durées de service *sont supposées indépendantes et identiquement distribuées (i.i.d.)*
- Elles sont caractérisées par leur loi de probabilité.

❖ Cas particulier important :

- durée de service exponentielle de paramètre μ ;
- durée moyenne de service : $t_s = 1/\mu$ (unité : seconde).

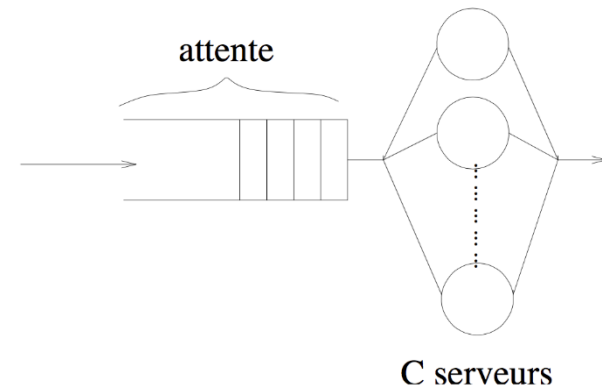
Nombre de serveurs

❖ Système Mono-serveur :



- Quand le serveur se libère, un des clients en attente prend le serveur (si la file d'attente n'est pas vide)

❖ Système Multiserveur :

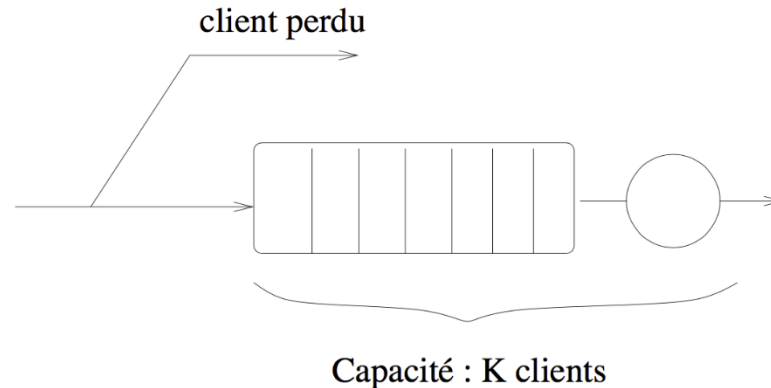


- Dès qu'un des serveurs se libère, un des clients en attente prend ce serveur (si la file d'attente est non vide)
- La loi du temps de service est en général supposée être la même pour chaque serveur

Nombre maximum des clients

❖ Nombre maximum de clients dans le système :

- nombre de serveurs + nombre de positions dans le buffer d'attente



❖ File d'attente finie :

- Si un client se présente et que la file est pleine, ce client est perdu
- Système à perte (perte pure, ou attente et perte)
- Le système est toujours stable par définition

❖ File d'attente infinie :

- Système sans perte par définition
- **Système à attente pure**
- Condition de stabilité : $\lambda < \mu$ (taux d'arrivée < vitesse de service)

Politiques de service

- ❖ Ordre dans lequel les clients accèdent aux ressources
- ❖ Discipline la plus classique : **FIFO** (First In First Out)
- ❖ Autres politiques de service :
 - LIFO = Last In First Out
 - *Random* = ordre aléatoire
 - Round Robin = un slot de temps successivement à chaque client
 - ***Processor Sharing***
 - partage équitable des ressources
 - s'il y a n clients dans le système simultanément ...
 - ... et si le serveur est capable de servir à un taux μ au maximum ...
 - ... alors les n clients sont servis simultanément, chacun au taux μ/n
 - *Processor Sharing* = cas limite du *Round Robin*

Notation de Kendall

$A/S/C/K/Z$

A : distribution des inter-arrivées

S : distribution des services

C : nombre de serveurs

K : capacité du système : nombre de serveurs + longueur de la file d'attente discipline de service

Z : discipline de service

ou A et S sont donnés par :

- M distribution exponentielle (Markov)
- G loi générale
- E_k loi de Erlang-k

Valeurs par défaut :

- K omis : la capacité de la file est infinie
- Z omis : discipline de service FIFO

Exemples :

- M/M/C : arrivées poissonniennes, services exponentiels, C serveurs, capacité infinie, FIFO
- M/M/C/C : arrivées poissonniennes, services exponentiels, C serveurs, pas de file d'attente, FIFO
- M/M/1/K : arrivées poissonniennes, services exponentiels, 1 serveur, capacité K, FIFO

Système à attente pure: M/M/1

Performances moyennes (dans le régime stationnaire)

Condition de stabilité : $\lambda < \mu$
(taux d'arrivée < vitesse de service du serveur)

Trafic offert ou facteur de charge (ρ) :

$$\rho = \lambda / \mu$$

Taux moyen d'utilisation : proportion du temps pendant lequel le serveur est actif (système non vide)

$$u = \rho = \frac{t_s}{t_a} = \frac{\lambda}{\mu}$$

Formule de Little :

$$N = \lambda T$$

où

- ❖ λ : nombre moyen de clients entrant dans le système par unité de temps
- ❖ T : temps moyen de séjour dans le système (=temps d'attente dans la file + temps de service)
- ❖ N : nombre moyen de clients dans le système.

Système à attente pure: M/M/1

Performances moyennes (dans le régime stationnaire)

Probabilité d'attendre : *probabilité d'arriver et de ne trouver aucun serveur libre* (P_A) (LOI D'ERLANG-C):

$$P_A = \rho = \frac{\lambda}{\mu}$$

Temps moyen de séjour dans le système (T): temps de service (t_s) + temps d'attente dans la file (W)

$$T = t_s + W = \frac{1}{\mu} + \frac{P_A}{(\mu - \lambda)}$$

Nombre moyen des clients dans le système (N) : nombre moyen des clients dans le serveur (N_s) + nombre moyen des clients dans la file (N_w)

$$N = N_s + N_w = \rho + \frac{\rho^2}{(1 - \rho)}$$

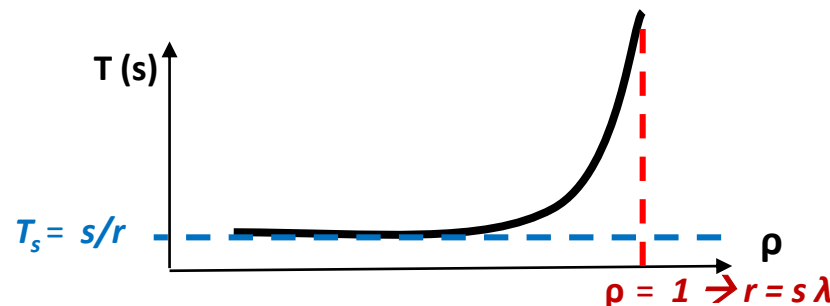
Buffer de sortie : M/M/1

Le lien de sortie d'un nœud (avec son buffer) peut être modélisé idéalement comme une file M/M/1

- ❖ Les paquets de données correspondent aux clients.
- ❖ Le serveur corresponde au lien de transmission avec un débit r (bps) par lequel des paquets de données sont transmis
- ❖ Le temps de service moyen (t_s) correspond à la durée de transmission (d_{trans}) sur le lien de sortie : $t_s = s$ (bits) / r (bps) = $1 / \mu$
- ❖ Le file d'attente est infinie.

Temps moyen de séjour dans le système (T): temps de service (t_s) + temps d'attente dans la file (W)

$$T = t_s + W = \frac{1}{\mu} + \frac{\rho}{(\mu - \lambda)} = \frac{1}{\mu} + \frac{1}{\mu} \frac{\lambda}{(\mu - \lambda)} = \frac{s}{r} + \frac{s}{r} \frac{s \lambda}{(r - s \lambda)} = d_{trans} + d_{buffer}$$

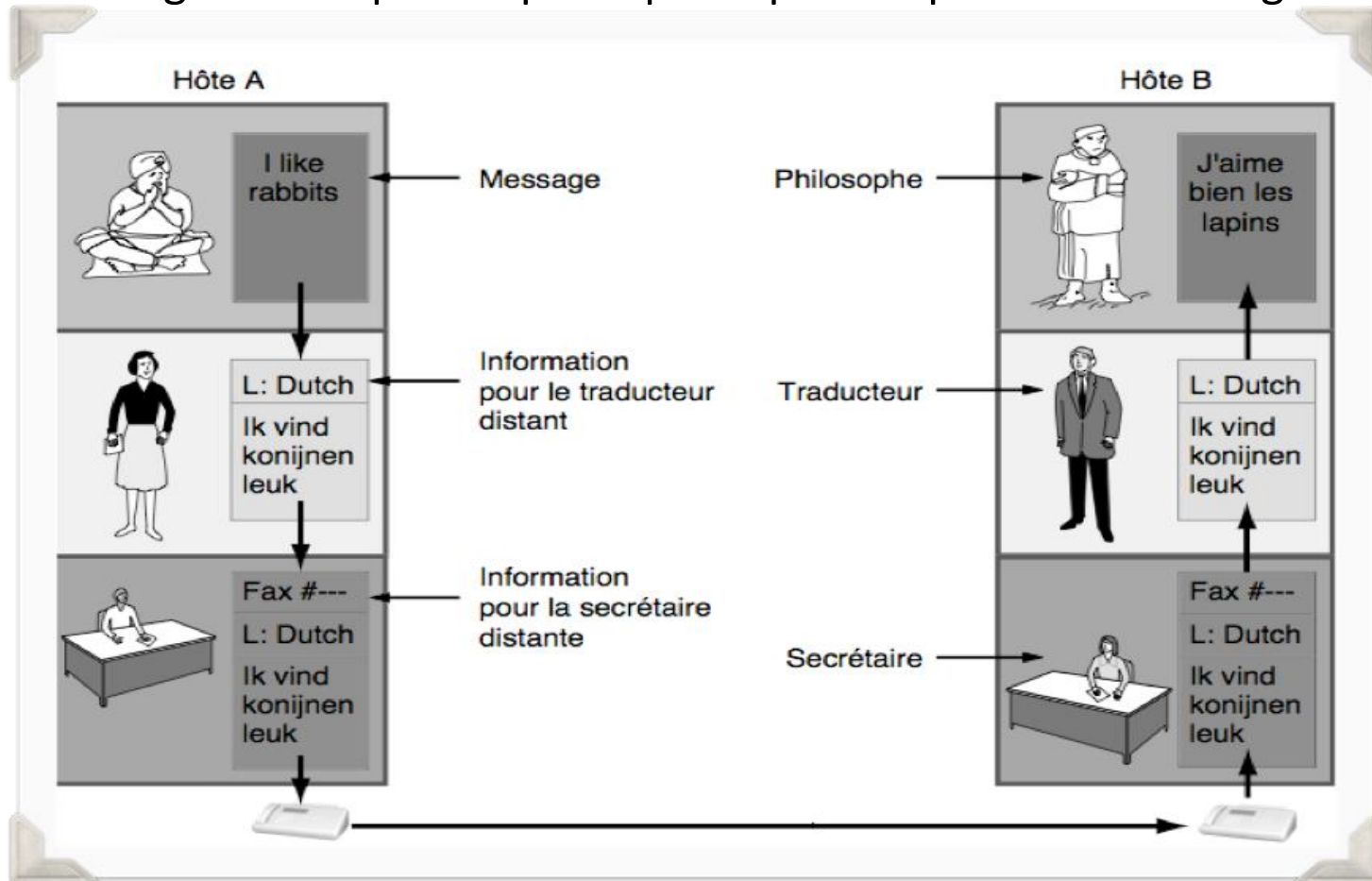


Qualité de Service (Quality of Service, QoS)

- ❖ Capacité à véhiculer dans de bonnes conditions un type de trafic (*Type of Service, ToS*) donné en termes de
 1. disponibilité
 2. latence
 3. variabilité de la latence (gigue)
 4. taux de perte de paquets
- ❖ But: optimiser les ressources d'un réseau
- ❖ Ensemble de facteurs de qualité nécessaires aux besoins d'une application
 - déf. générique : à préciser cas par cas
 - dépend de la nature des application (vidéo, audio, email, transfert de fichiers)
- ❖ Exemples
 - Faible taux d'erreur (proba perte/modif. message)
 - nécessaire si les données sont peu redondantes
 - Stabilité latence (gigue petite)
 - surtout pour applications multimédia
 - Garantie limite supérieur sur latence
 - appli critiques temps réel

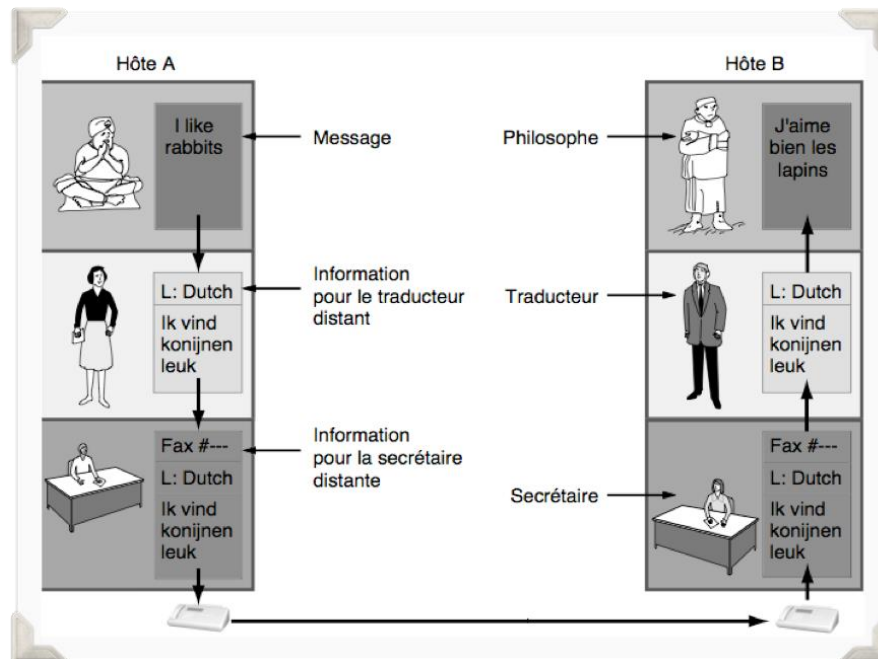
Protocoles des réseaux

- ❖ Analogie : deux philosophes qui ne parlent pas la même langue



Protocoles des réseaux

- ❖ Suite d'étapes (tâches) regroupées en ***une structure de couches***
- ❖ ***Chaque couche*** met en œuvre un service :
 - Via ses propres fonctions internes de couche
 - En s'appuyant sur les services (fonctions) fournis par la couche inférieure

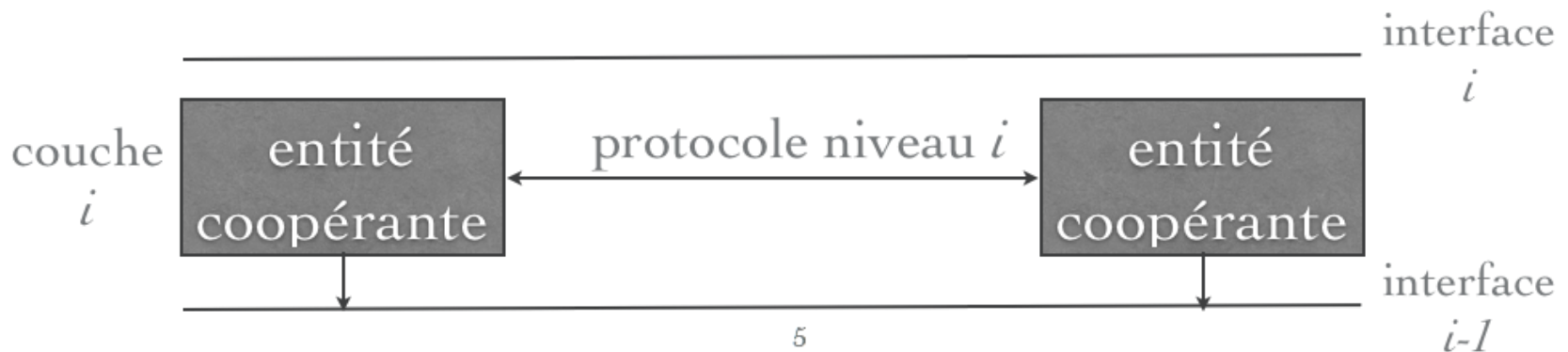


Protocoles des réseaux

- ❖ Couches, niveaux, strates..., **pourquoi ?**
 - gage de modularité
 - facilite la maintenance
- ❖ Stratégie
 - matériel et logiciel considérés de la même façon
 - forte structuration des logiciels réseau
- ❖ Mécanisme virtuel où chaque couche
 - fournit un service à la couche immédiatement supérieure
 - dissimule les détails d'implémentation
- ❖ *Protocole* et interface implémentent ce fonctionnement en couches; le *service* offert correspond à la sémantique de la couche

Protocoles des réseaux

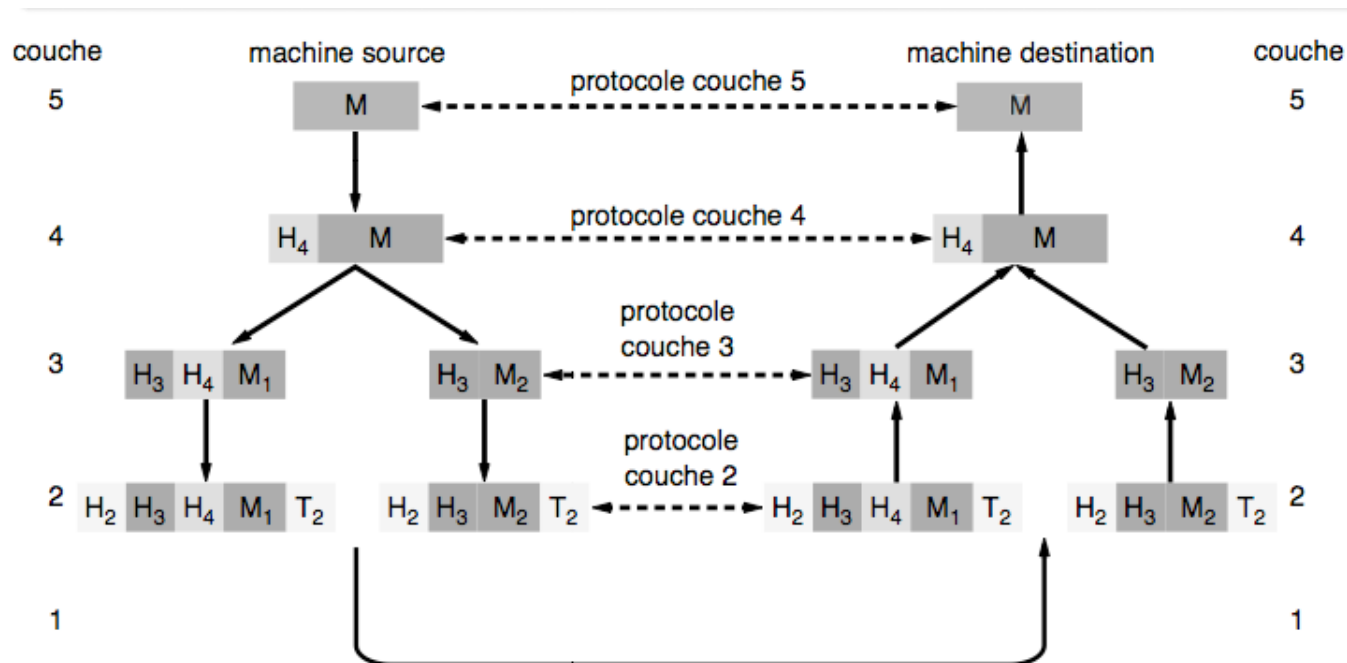
- ❖ **Communication**: entre couches de même niveau
- ❖ **Interface** : ensemble des fonctions (logicielles et matérielles) et des règles d'accès au service de la couche inférieure.
- ❖ **Protocole** : ensemble des conventions nécessaires pour faire coopérer des entités distantes, en particulier pour établir et entretenir des échanges d'informations entre ces entités.
- ❖ La conception protocole utilise souvent le protocole de niveau inférieur (plus élémentaire), au moyen de son interface
- ❖ **Protocol Data Unit** ou *Unité de données de protocole (PDU)* : est *l'unité minimale* des informations échangées *entre deux entités au niveau de la même couche*



Protocoles des réseaux

❖ Communication virtuelle vs physique

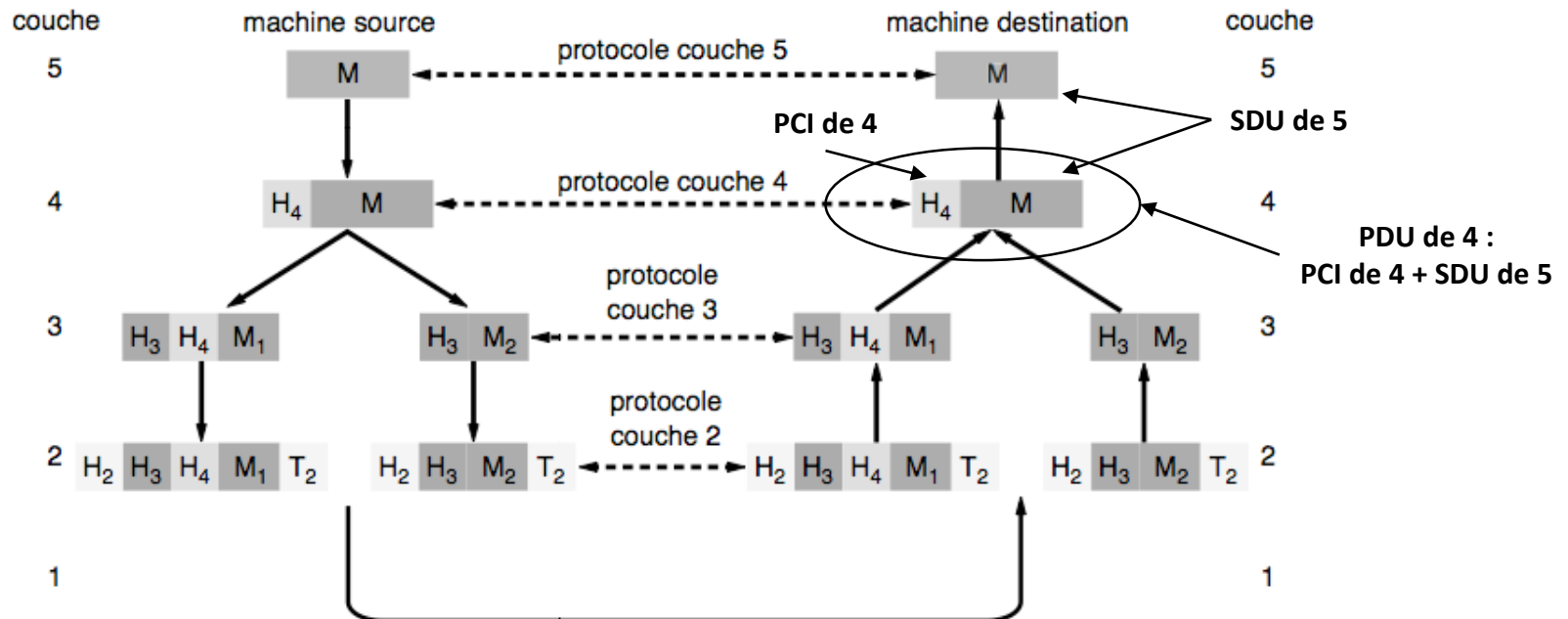
- La *communication* entre couches du *même niveau* est *virtuelle* (passage de PDUs)
- Pas de communication physique** entre couches du *même niveau*
 - aucune donnée transmise **physiquement** d'une couche *i* à une autre couche *i*



Protocoles des réseaux

❖ Encapsulation

- **Service Data Unit** ou *Unité de données de service (SDU)* : est *l'unité minimale* des informations échangées **entre deux couches** dans la même entité. Ce sont les *messages M* proprement dits, la donnée effective ou *payload (charge utile)*
- **Protocol-Control information (PCI)** : En-têtes (*headers H*) avec une information de contrôle dépendant du protocole.
- Dans une couche *N*, on **encapsule** un SDU de la couche *N+1* en l'ajoutant le PCI de *N*. Le résultat est le PDU de la couche *N*.



Protocoles des réseaux

❖ Spécification

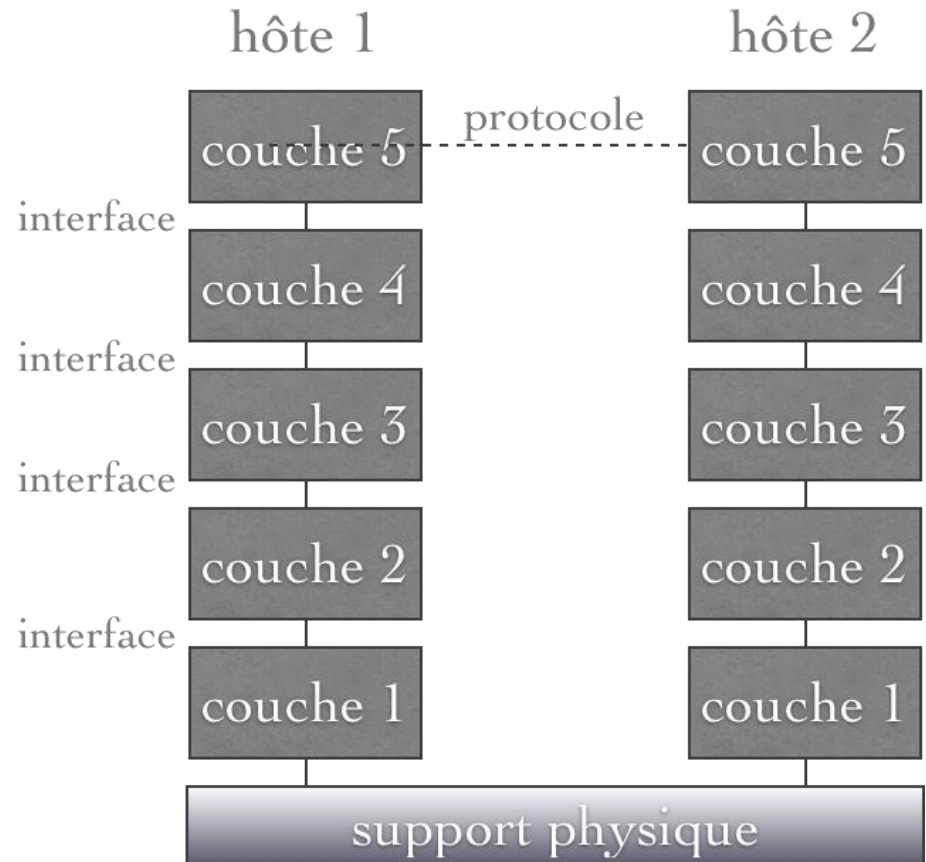
- format et ordonnancement messages échangés entre ≥ 2 entités de comm.
- actions à réaliser lors de l'émission/réception msg

❖ Ensemble de règles

- de comm. (langue commune)
- de bon fonctionnement (partage ressource)

❖ pour

- utiliser le support physique
- transporter/utiliser l'info



Pile de protocoles OSI

❖ Open Systems Interconnection

- Proposition d'architecture réseaux différente selon le constructeur
- Interconnexion difficile due au caractère propriétaire
- Intervention ISO
 - but: normalisation protocoles
- Principes:
 - une couche par niveau d'abstraction
 - fonctions bien définies par couche
 - nombre de couches raisonnable
- Résultat: pile de protocoles OSI
 - Sept couches



Pile de protocoles OSI

❖ Physique (1):

- **concerne la transmission et la réception du flux de bits non structurés sur le support physique**
- définit les moyens de transmettre des bits
- définit caractéristiques optiques, électriques ou mécaniques

❖ Liaison de données ou de lien (2):

- **permet un transfert sans erreur des trames de données d'un nœud à un autre sur la couche physique**
- définit les procédures d'exploitation du lien de comm
- permet l'envoi de trames en séquence
- détecte et corrige les erreurs du support physique
- contrôle l'accès au canal partagé (sous-couche MAC)



Pile de protocoles OSI

❖ Réseau (3):

- **contrôle les opérations du sous-réseau décidant quel chemin physique prend les données**
- détermine comment les données traversent le sous-réseau
- route les paquets en fonction d'adresses réseau uniques
- gère les passages entre différents réseaux

❖ Transport (4):

- **assure que les messages sont délivrés sans erreur, dans l'ordre, et sans pertes ni duplications**
- assure un transfert fiable et en séquence des messages
- permet le découpage / réassemblage des données
- permet un contrôle de congestion



Pile de protocoles OSI

❖ Session (5):

- permet l'établissement de session entre des processus s'exécutant sur différentes stations
- assure la gestion du dialogue et du jeton
- gère la synchronisation

❖ Présentation (6):

- formats les données à présenter à la couche application.
- peut être considéré comme le « traducteur » du réseau
- définit la syntaxe et la sémantique des données
- gère conversion, reformate, comprime, chiffre

❖ Applications (7):

- sert de fenêtre pour les utilisateurs et le processus d'application pour accéder aux services réseau
- assure les services de base: (Telnet, http, smtp,....)



Pile de protocoles OSI

OSI (Open Source Interconnection) 7 Layer Model

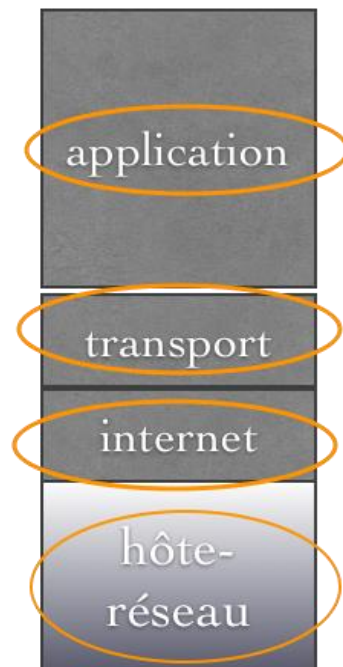
Layer	Application/Example		Central Device/ Protocols		DOD4 Model
Application (7) Serves as the window for users and application processes to access the network services.	End User layer Program that opens what was sent or creates what is to be sent Resource sharing • Remote file access • Remote printer access • Directory services • Network management		User Applications SMTP		G A T E W A Y Can be used on all layers
Presentation (6) Formats the data to be presented to the Application layer. It can be viewed as the "Translator" for the network.	Syntax layer encrypt & decrypt (if needed) Character code translation • Data conversion • Data compression • Data encryption • Character Set Translation		JPEG/ASCII EBDIC/TIFF/GIF PICT		
Session (5) Allows session establishment between processes running on different stations.	Synch & send to ports (logical ports) Session establishment, maintenance and termination • Session support - perform security, name recognition, logging, etc.		Logical Ports RPC/SQL/NFS NetBIOS names		
Transport (4) Ensures that messages are delivered error-free, in sequence, and with no losses or duplications.	TCP Host to Host, Flow Control Message segmentation • Message acknowledgement • Message traffic control • Session multiplexing	F I L T E R I N G	TCP/SPX/UDP		Host to Host
Network (3) Controls the operations of the subnet, deciding which physical path the data takes.	Packets ("letter", contains IP address) Routing • Subnet traffic control • Frame fragmentation • Logical-physical address mapping • Subnet usage accounting		Routers IP/IPX/ICMP		Internet
Data Link (2) Provides error-free transfer of data frames from one node to another over the Physical layer.	Frames ("envelopes", contains MAC address) [NIC card — Switch — NIC card] (end to end) Establishes & terminates the logical link between nodes • Frame traffic control • Frame sequencing • Frame acknowledgment • Frame delimiting • Frame error checking • Media access control		Switch Bridge WAP PPP/SLIP	Land Based Layers	Network
Physical (1) Concerned with the transmission and reception of the unstructured raw bit stream over the physical medium.	Physical structure Cables, hubs, etc. Data Encoding • Physical medium attachment • Transmission technique - Baseband or Broadband • Physical medium transmission Bits & Volts		Hub		

Modèle TCP/IP vs OSI

- ❖ Modèle OSI tout à fait valable, reconnu et étudié, mais incapable de s'imposer:
 - mauvaise chronologie/technologie/implémentation
- ❖ Reste une référence, pas une technologie. D'où le modèle TCP/IP

OSI

TCP/IP

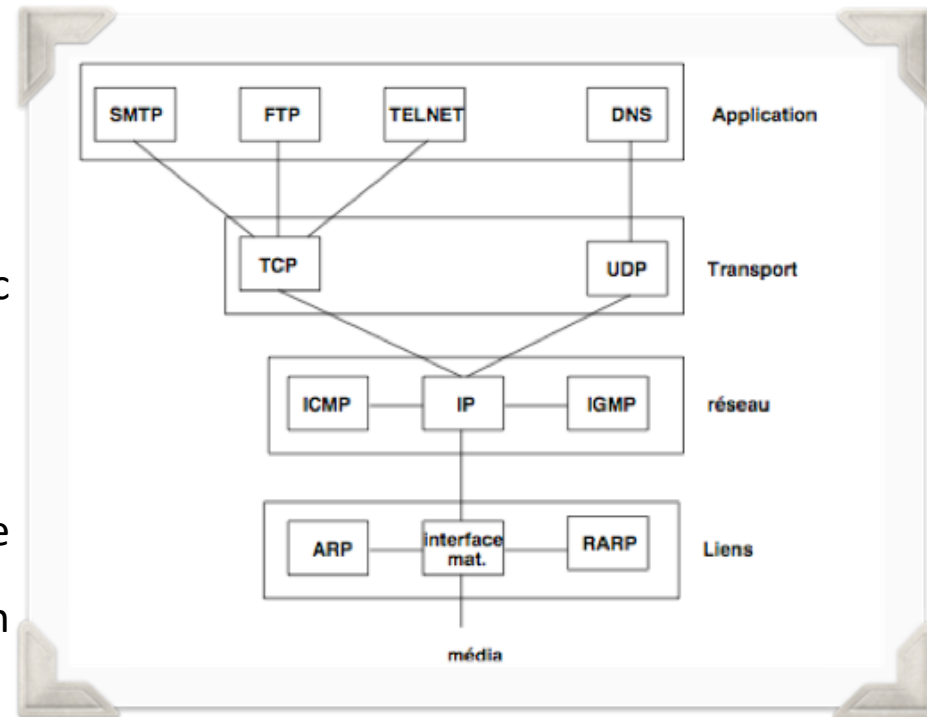


offre les fonctions et services des couches supérieures

- fournit transmission fiable avec connexion TCP
- fournit transmission fiable sans connexion UDP
- assure contrôle de flux
- interconnecte rx distants sans connexion
- route paquets IP sur tout le rx
- cherche à éviter congestions
- pas vraiment spécifiée
- assure envoi paquets IP
- implémentation liée techno du LAN

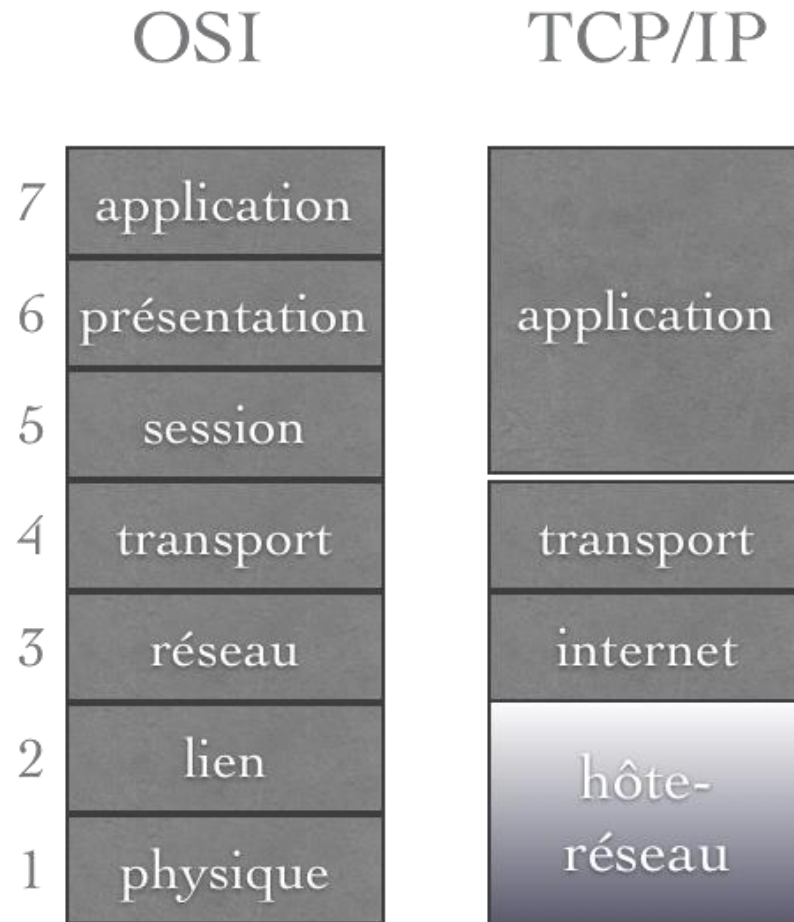
Protocoles Principaux

- ❖ Application :
 - FTP : transfert fichiers
 - SMTP : messagerie électronique
 - TELNET : présentation d'écran
 - DNS : résolution IP - nom de domaine
- ❖ Transport :
 - TCP : assure service fiable avec connexion
 - UDP : service pas fiable sans connexion
- ❖ Réseaux:
 - IP : assure service sans connexion
 - ICMP : envoi de messages entre couches IP de \neq nœuds
 - IGMP : gestion appartenance à un groupe Xcast
- ❖ Liens :
 - ARP : résolution IP - MAC
 - Interface matériel : Ethernet



PDU selon couche

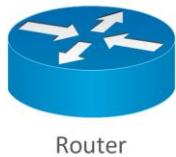
- ❖ Application :
 - Message
- ❖ Transport :
 - TCP : Segment
 - UDP : Datagramme
- ❖ Réseaux:
 - Paquet
- ❖ Liens :
 - Trame (*frame*)
- ❖ Physique (média)
 - Bit



Noeud (dispositif) selon couche

❖ Réseaux:

- Routeur (*router*)



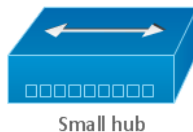
❖ Liens :

- Commutateur (*switch*)



❖ Physique (média)

- Hub



OSI

TCP/IP

