



# Partie 6 : Couche réseau et interconnexion de réseaux

Olivier GLÜCK

Université LYON 1 / Département Informatique

Olivier.Gluck@univ-lyon1.fr

<http://perso.univ-lyon1.fr/olivier.gluck>



# Copyright

---

- Copyright © 2023 Olivier Glück; all rights reserved
- Ce support de cours est soumis aux droits d'auteur et n'est donc pas dans le domaine public. Sa reproduction est cependant autorisée à condition de respecter les conditions suivantes :
  - Si ce document est reproduit pour les besoins personnels du reproducteur, toute forme de reproduction (totale ou partielle) est autorisée à la condition de citer l'auteur.
  - Si ce document est reproduit dans le but d'être distribué à des tierces personnes, il devra être reproduit dans son intégralité sans aucune modification. Cette notice de copyright devra donc être présente. De plus, il ne devra pas être vendu.
  - Cependant, dans le seul cas d'un enseignement gratuit, une participation aux frais de reproduction pourra être demandée, mais elle ne pourra être supérieure au prix du papier et de l'encre composant le document.
  - Toute reproduction sortant du cadre précisé ci-dessus est interdite sans accord préalable écrit de l'auteur.



# Remerciements

---

- Certains transparents sont basés sur des supports de cours de :
  - Danièle DROMARD (PARIS 6)
  - Andrzej DUDA (INP Grenoble/ENSIMAG)
  - Shivkumar KALYANARAMAN (RPI/ECSE)
  - Alain MILLE (LYON 1)
  - CongDuc PHAM (LYON 1)
  - Laurent Toutain (ENST Bretagne)
  - Michel RIVEILL (Université de Nice/ESSI)
  - l' Institut National des Télécommunications (INT)
- Des figures sont issues des livres cités en bibliographie



# Bibliographie

---

- « *Réseaux* », 4ième édition, Andrew Tanenbaum, Pearson Education, ISBN 2-7440-7001-7
- « *Réseaux et Télécoms* », Claude Servin, Dunod, ISBN 2-10-007986-7
- « *Analyse structurée des réseaux* », 2ième édition, J. Kurose et K. Ross, Pearson Education, ISBN 2-7440-7000-9
- « *TCP/IP Illustrated Volume 1, The Protocols* », W. R. Stevens, Addison Wesley, ISBN 0-201-63346-9
- « *TCP/IP, Architecture, protocoles, applications* », 4ième édition, D. Comer, Dunod, ISBN 2-10-008181-0
- « *An Engineering Approach to Computer Networking* », Addison-Wesley, ISBN 0-201-63442-6
- Internet...
  - <http://www.guill.net/>
  - <http://www.courseforge.org/courses/>
  - <http://www.commentcamarche.net/ccmdoc/>
  - <http://www.rfc-editor.org/> (documents normatifs dans TCP/IP)



# Bibliographie

- « Réseaux », 4ième édition, Andrew Tanenbaum, Pearson Education, ISBN 2-7440-7001-7
- « Réseaux et Télécoms », Claude Servin, Dunod, ISBN 2-10-007986-7
- « Réseaux locaux et Internet, des protocoles à l'interconnexion », 3ième édition, Laurent Toutain, Hermes Science, ISBN 2-7462-0670-6
- « An Engineering Approach to Computer Networking », Addison-Wesley, ISBN 0-201-63442-6
- Internet...
  - <http://www.guill.net/>
  - <http://www.courseforge.org/courses/>
  - <http://www.commentcamarche.net/ccmdoc/>
  - <http://www.protocols.com/>
  - [http://dir.yahoo.com/Computers\\_and\\_Internet/](http://dir.yahoo.com/Computers_and_Internet/)
  - <http://www.rfc-editor.org/> (documents normatifs dans TCP/IP)



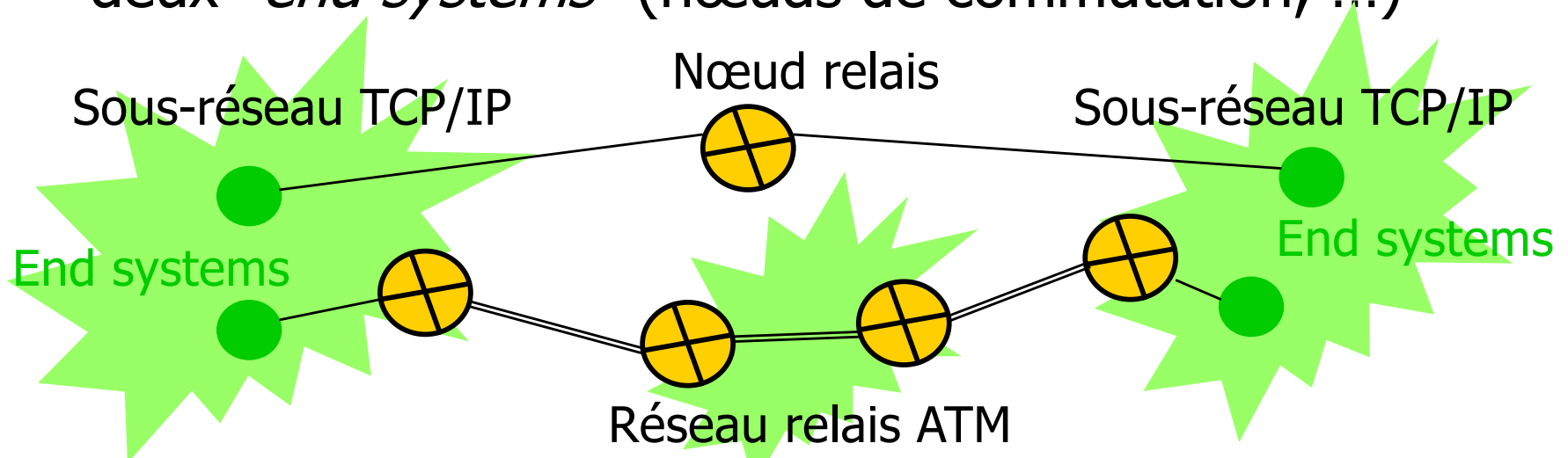
# Introduction (1)

---

- Pour échanger des informations entre 2 entités communicantes quelconques à travers un ou plusieurs réseaux :
  - les deux correspondants doivent être mis en relation (**notion de commutation**)
  - chaque correspondant doit être identifié et localisé de manière unique sur le réseau (**notions d'adressage et de nommage**)
  - le réseau doit acheminer les blocs d'information vers le destinataire (**notion de routage**)
  - la taille des unités de données transférées doit être adaptée aux capacités du réseau (**notion de segmentation**)

# Introduction (2)

- le trafic admis dans le réseau ne doit pas conduire à l'effondrement de celui-ci (**notion de contrôle de congestion**)
- On distingue
  - les nœuds supportant les applications utilisateurs (hôtes ou "*end systems*")
  - les nœuds servant de relais à la communication entre deux "*end systems*" (nœuds de commutation, ...)





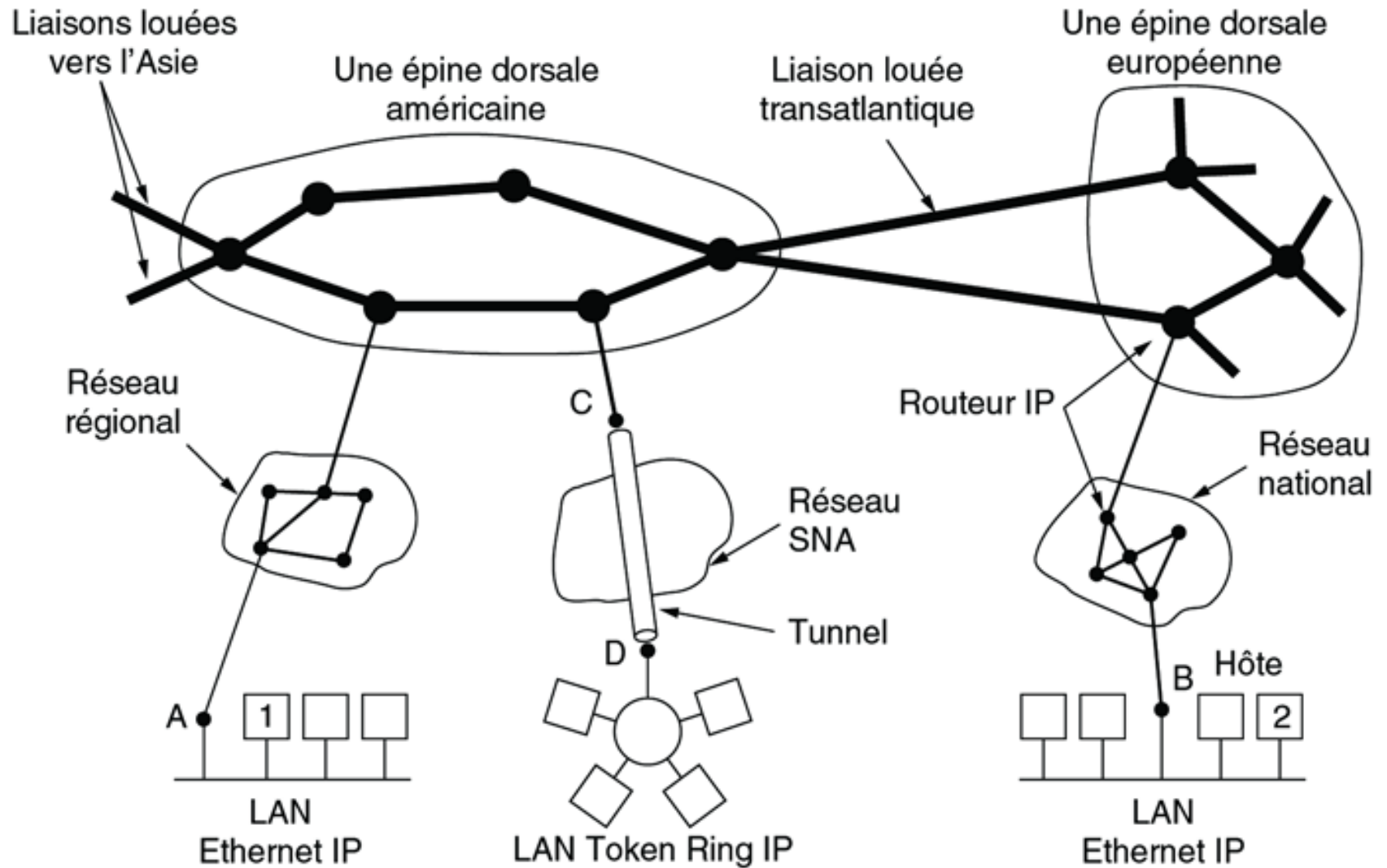
# Introduction (3)

---

- Problèmes liés à l'interconnexion des réseaux et aux réseaux longues distances
  - grande couverture géographique
  - hétérogénéité
    - des réseaux raccordés (clients)
    - des modes de transmission (cœur et clients)
    - des applications
  - agrégation de trafic importante (risque de congestion)
  - mélange de réseaux publics et privés
  - tarification du trafic par les opérateurs avec éventuellement des qualités de services différentes
  - soucis de performance (traversée des équipements) et d'équité (entre clients)



# Introduction (4)



© Pearson Education France



# Plan de la partie 6

---

- Commutation et multiplexage dans les réseaux
- Notion d'adressage et de nommage
- L'acheminement dans le réseau
- Adaptation de la taille des unités de données
- La congestion dans les réseaux
- Les équipements d'interconnexion



# Commutation et multiplexage

---



# Pourquoi multiplexer et commuter ?

---

- Quel est le problème ?
  - Comment mettre en relation deux entités communicantes du réseau ?
    - interconnexion totale de N stations =  $N(N-1)/2$  liens physiques
  - Comment répartir la charge du réseau et être résistant aux pannes ?
    - résistance aux pannes : plusieurs chemins pour aller de A à B
    - si plusieurs chemins, lequel choisir ?
- Réseau maillé : deux stations clientes du réseau peuvent être mises en relation par différents chemins



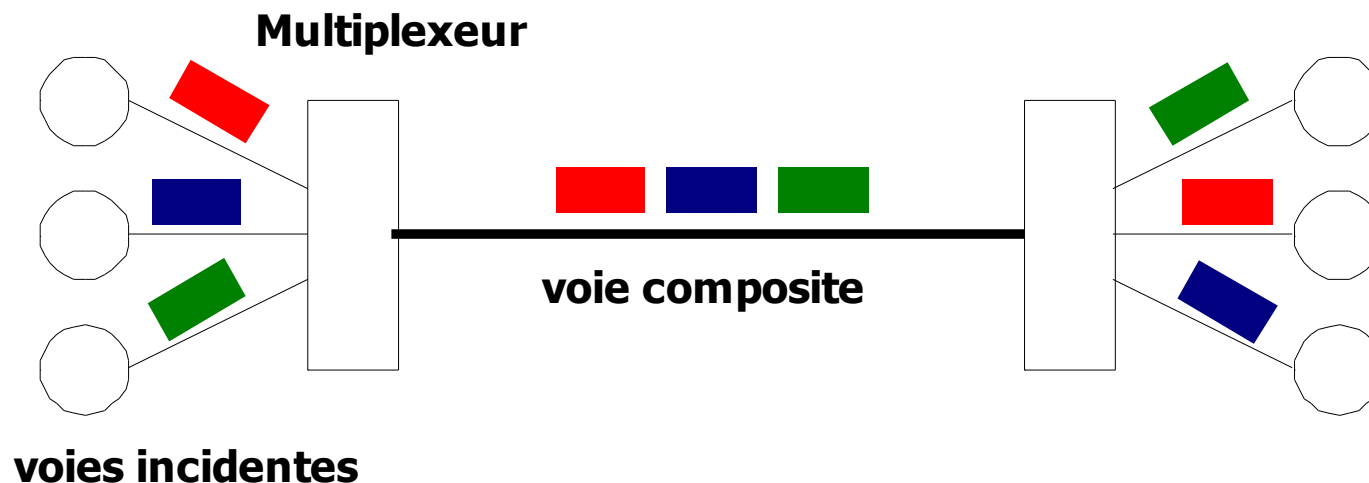
# Pourquoi multiplexer et commuter ?

---

- Minimiser le coût de l'infrastructure du réseau
  - multiplexage de plusieurs communications sur un même lien physique
  - minimisation du nombre de liens au sein du réseau
- Optimiser l'infrastructure du réseau
  - le nombre de communications potentielles peut être très supérieur au nombre de liens du réseau de commutation
  - atteindre le destinataire de la façon la plus efficace
  - optimiser l'utilisation des ressources du réseau : si chaque abonné désire joindre un autre abonné, il faut optimiser le partage des ressources

# Principe du multiplexage

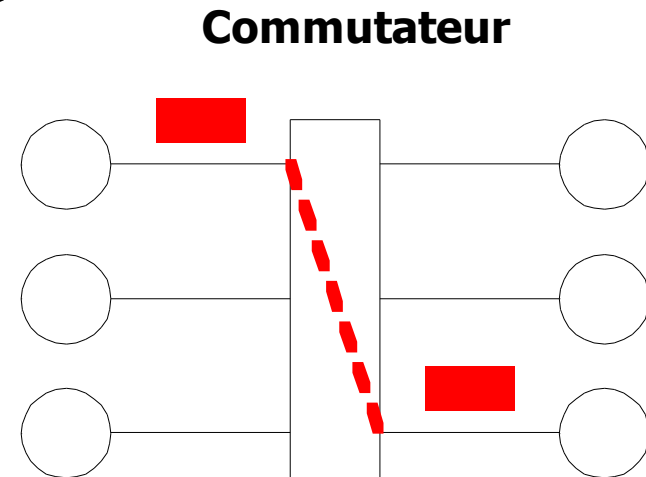
- Acheminement simultané de plusieurs communications sur un même lien physique



- Multiplexage : regroupement de plusieurs voies incidentes sur un même support
- Démultiplexage : restitution à chaque destinataire des données des différentes voies

# Principe de la commutation

- Aiguillage de la communication d'un canal en entrée vers un canal de sortie
- Diverses techniques
  - commutation de circuits
  - commutation de messages
  - commutation de paquets
  - commutation de trames
  - commutation de cellules
- Un réseau à commutation assure une connectivité totale du point de vue utilisateurs (différence entre topologie physique et topologie logique)





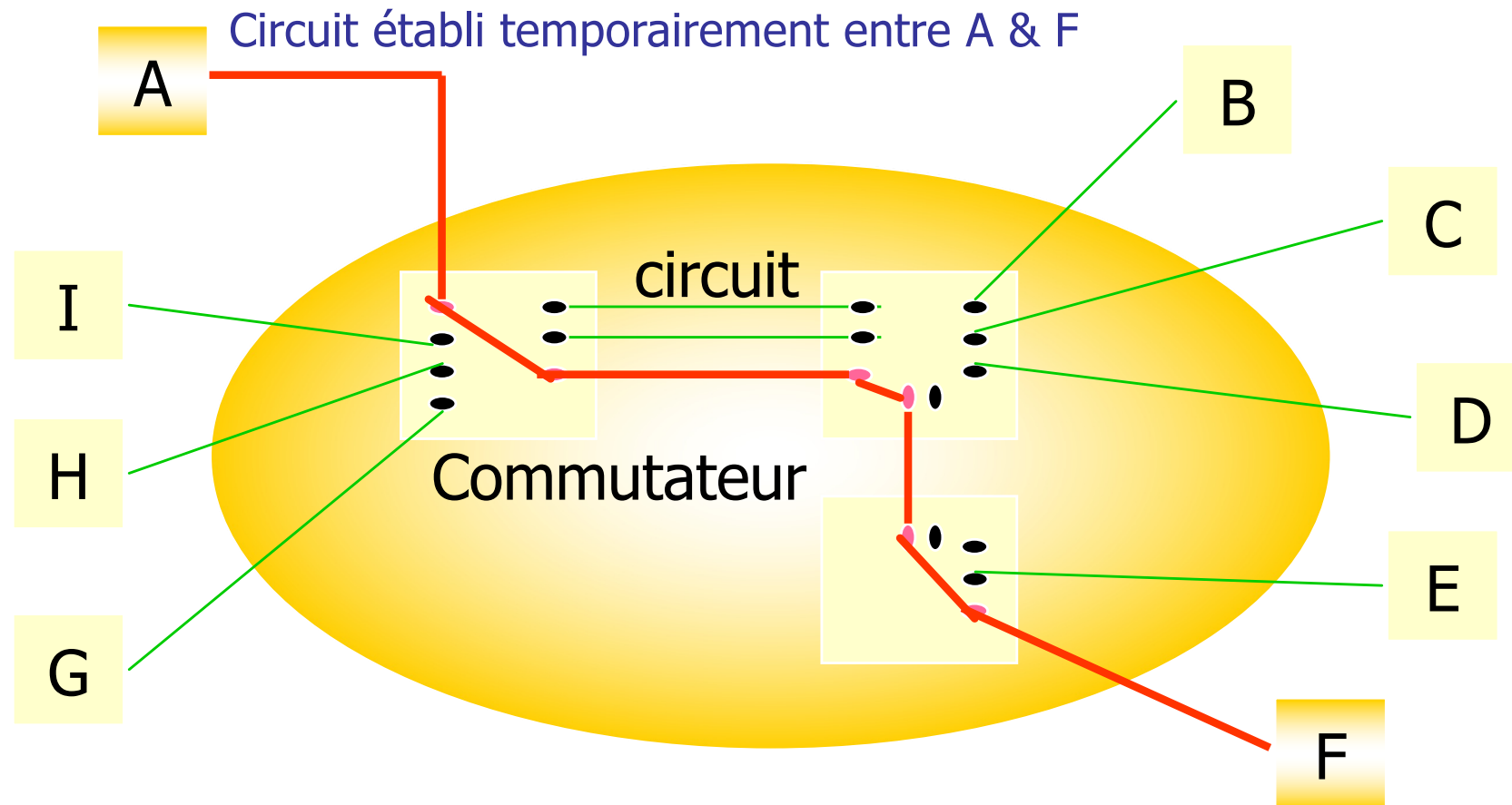
# La commutation de circuits (1)

---

- La commutation la plus simple
- Un chemin physique est constitué de bout en bout entre une source et une destination
  - juxtaposition de différents supports physiques => circuit établi entre les deux abonnés
  - la mise en relation est réalisée par les commutateurs avant tout échange de données (phase de connexion)
  - le circuit est bloqué (phase de transfert) tant que les deux abonnés ne le restituent pas explicitement (phase de libération)
- Exemple : le Réseau Téléphonique Commuté



# La commutation de circuits (2)





# La commutation de circuits (3)

---

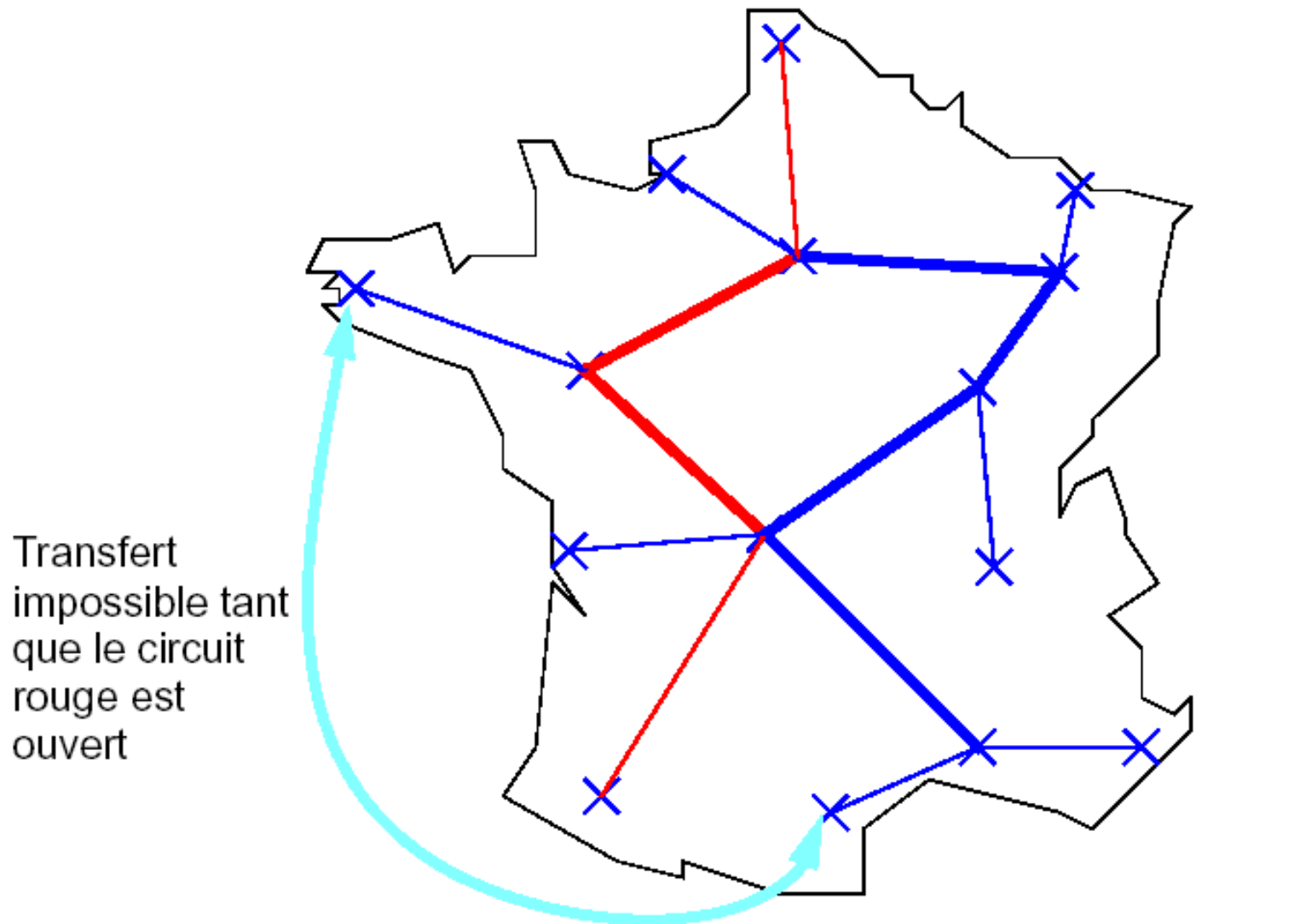
## ■ Caractéristiques

- garantit le bon ordonnancement des données
- pas de stockage intermédiaire des données
- débits source/destinataire identiques
- les abonnés monopolisent la ressource durant toute la connexion
- facturation à la minute

## ■ Inconvénients

- s'il n'y a plus de ressource disponible de bout en bout, la connexion est refusée
- mauvaise utilisation des ressources : les deux abonnés consomment rarement toute la bande passante

# La commutation de circuits (4)



source L. Toutain



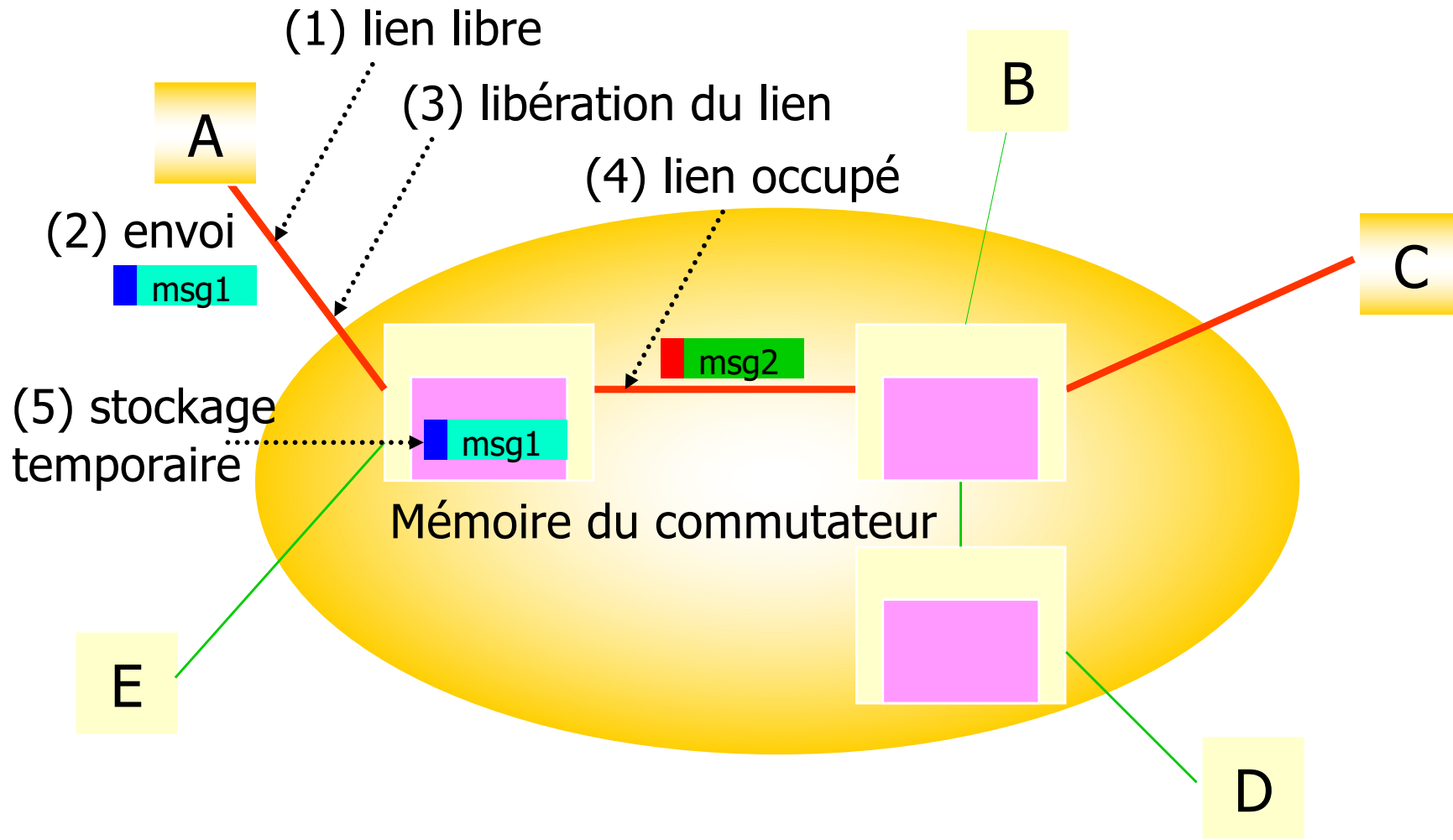
# La commutation de messages (1)

---

## ■ Principe

- pas d'établissement préalable de la communication (aucun lien physique entre la source et le destinataire)
- un message constitue une unité de transfert qui est acheminée individuellement dans le réseau (exemple : un fichier)
- sur chaque nœud du réseau, un message est
  - reçu en entier
  - stocké si nécessaire (occupation des lignes)
  - analysé (contrôle des erreurs)
  - transmis au nœud suivant, etc...
- facturation en fonction de la quantité de données

# La commutation de messages (2)





# La commutation de messages (3)

---

## ■ Avantages

- meilleure utilisation des liens qu'avec la commutation de circuit -> meilleur dimensionnement du réseau
- en cas de fort trafic, il n'y a pas de blocage lié au réseau empêchant l'émission : le message est simplement ralenti
- possibilité de faire de la diffusion d'un même message à plusieurs correspondants
- possibilité de faire du contrôle d'erreurs entre deux commutateurs voisins -> fiabilité

## ■ Inconvénients

- nécessite une mémoire de masse importante dans les commutateurs
- temps d'acheminement non maîtrisé
- si un message est corrompu, il devra être retransmis intégralement

## ■ Exemple d'application : systèmes de messagerie



# La commutation de paquets (1)

---

- Principe : + de souplesse
  - le message est découpé en paquets (fragments) de petite taille
  - chaque paquet est acheminé dans le réseau indépendamment du précédent (et des autres)
  - pas de stockage d'information dans les nœuds intermédiaires (réémission immédiate sur le lien optimal)
  - le séquençement des paquets n'est plus garanti
  - reconstitution du message à l'arrivée avec éventuellement un réordonnancement des paquets
  - multiplexage des paquets de différentes sources sur un même lien (liens banalisés et non pas dédiés)



# La commutation de paquets (2)

---

## ■ Avantages

- optimisation de l'utilisation des ressources, plus grande équité entre les utilisateurs
- transmission plus rapide que dans la commutation de messages
- retransmission uniquement du paquet erroné en cas d'erreurs

## ■ Inconvénients

- il peut être nécessaire de réordonner les paquets pour reconstituer le message
- chaque paquet doit contenir les informations nécessaires à son acheminement





# La commutation de paquets (3)

---

- Acheminement en mode datagramme ou mode non connecté
  - exemple : Internet
  - les paquets peuvent emprunter des chemins différents
  - ordre d'arrivée différent, reconstitution à l'arrivée
  - risque de perte de paquets si un nœud est engorgé (pas de contrôle de flux possible de bout en bout)
  - simplicité de mise en œuvre (la complexité est reportée dans les extrémités qui doivent reconstruire le message)
  - routage différent de chaque paquet permet de répartir la charge du réseau (routage adaptatif)



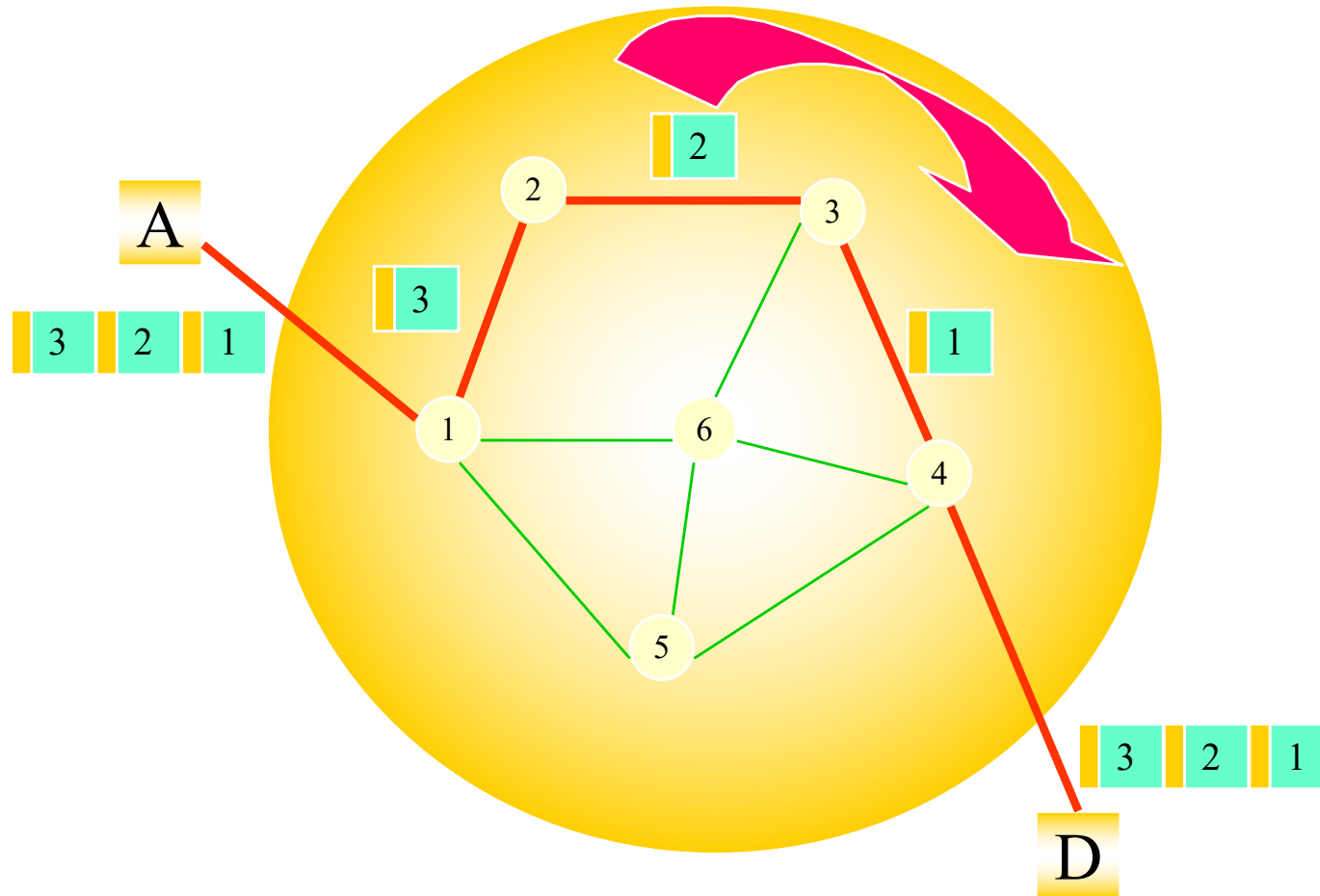
# La commutation de paquets (4)

---

- Acheminement en mode circuit virtuel ou mode connecté
  - exemple : Transpac
  - un chemin est établi à l'avance (le Circuit Virtuel)
  - les paquets sont reçus dans l'ordre (même chemin pour tous)
  - décision d'acheminement plus rapide (la route est connue)
  - réalisation plus difficile
  - compromis entre la commutation de circuits et la commutation de paquets en mode datagramme
    - optimisation des ressources
    - garantie de séquençement

# Le mode circuit virtuel

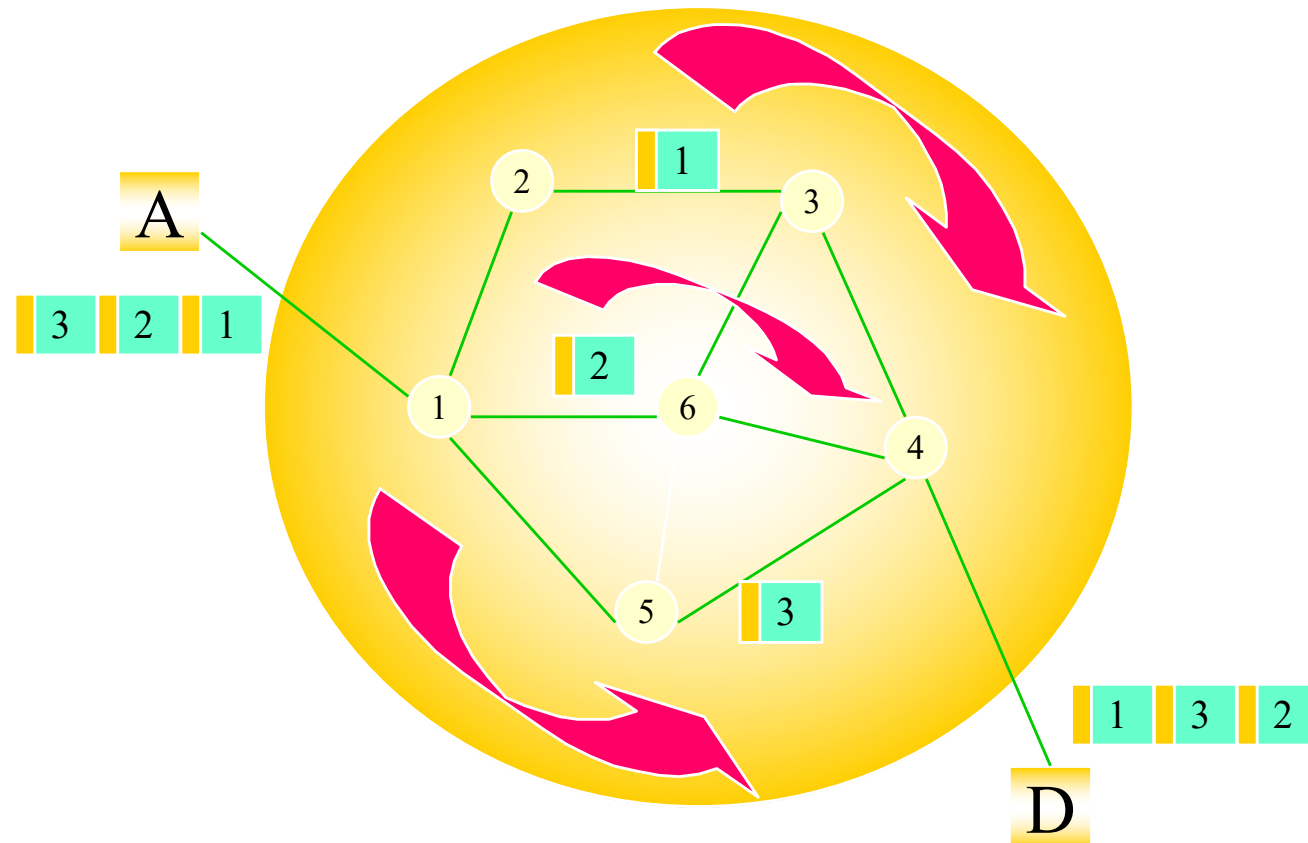
Exemple : X25



source L. Toutain

# Le mode datagramme

Exemple : IP (Internet)





# Circuit virtuel vs. Datagramme (1)

---

- Le mode Circuit Virtuel (ATM, Frame Relay)
  - avantages :
    - séquençement des paquets garanti
    - court en-tête, acheminement plus rapide
  - inconvénients :
    - chaque connexion ouverte consomme des ressources même si aucun paquet ne circule
    - une seule route possible
      - pas de contournement des zones congestionnées
      - moins robuste (défaillance de routeurs sur l'itinéraire)
    - perte de tous les circuits en cas de défaillance d'un routeur



# Circuit virtuel vs. Datagramme (2)

---

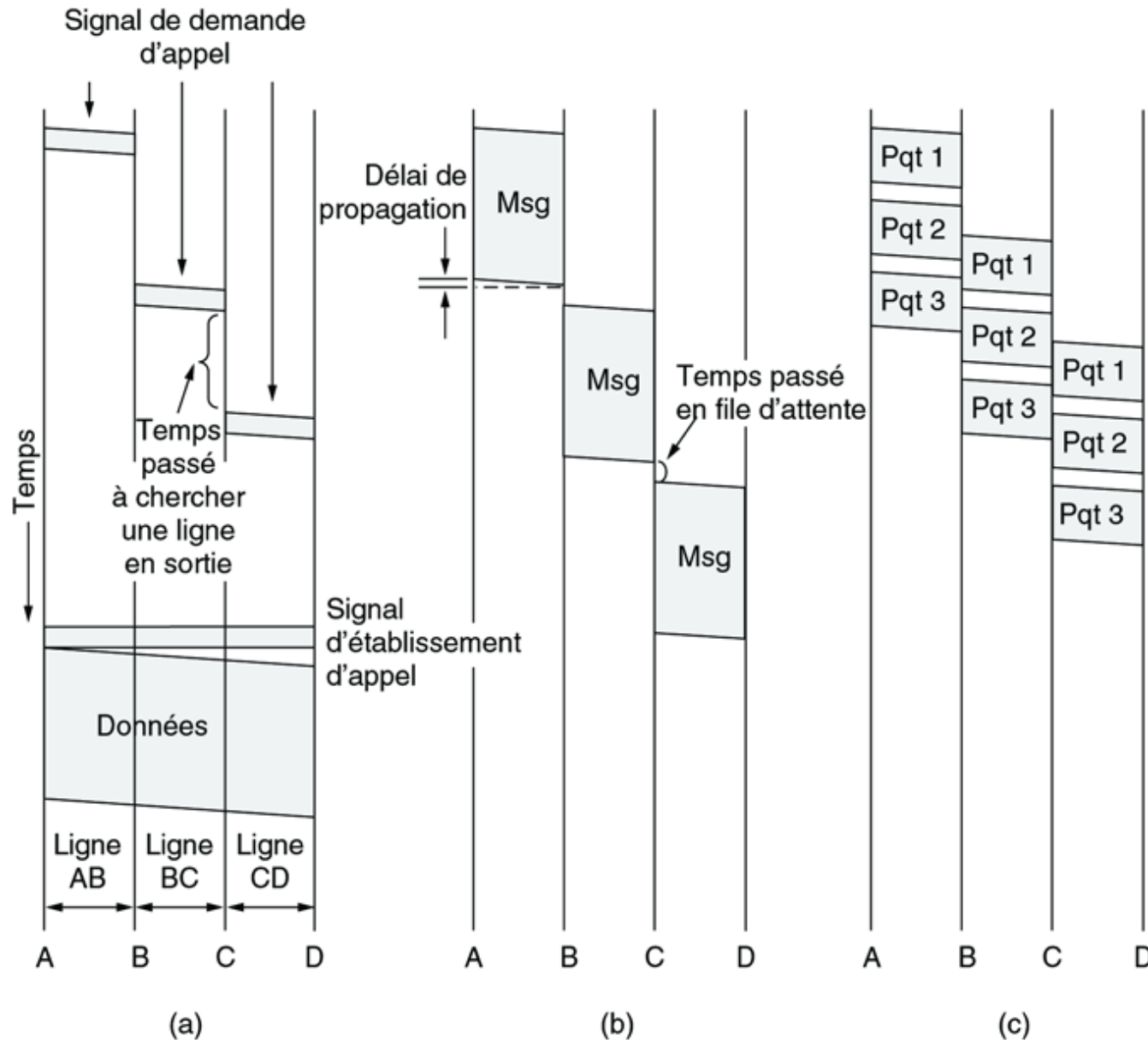
- Le mode Datagramme (Internet)
  - avantages :
    - plus robuste (plusieurs routes possibles)
      - résistance aux défaillances de routeur
      - en cas de crash d'un routeur, perte uniquement des paquets en cours de traitement
      - adaptabilité aux congestions
    - répartition de la charge
  - inconvénients :
    - risque accru de congestion
    - arrivée désordonnée des paquets
    - décision d'acheminement sur chaque paquet

# Commutation dans le temps

[http://wps.aw.com/aw\\_kurose\\_network\\_2/0,7240,227091-,00.html](http://wps.aw.com/aw_kurose_network_2/0,7240,227091-,00.html)



- Commutation de
- (a) circuits
- (b) messages
- (c) paquets





# Adressage et nommage

---





# Adressage et nommage - définitions

---

- Adressage = moyen d'identifier de manière unique deux entités communicantes
  - la source doit pouvoir fournir au réseau l'adresse du destinataire
  - le destinataire doit pouvoir identifier l'adresse de son correspondant (adresse source)
- Adressage physique et adressage logique
  - adressage physique : désigne le point de raccordement au réseau (localisation des extrémités)
  - adressage logique ou nommage : identifie un processus ou une machine indépendamment de sa localisation (désigne l'objet de la communication : ex. (@IP, port) des *sockets*)



# Les types d'adressage

---

- Adressage hiérarchique
  - l'adresse est décomposée en différentes parties qui permettent d'identifier
    - le réseau auquel l'utilisateur est rattaché
    - le point d'accès par lequel il est raccordé au réseau
    - l'utilisateur dans l'installation locale
  - le champ adresse diminue au fur et à mesure de la progression des blocs dans le réseau
  - exemple : numérotation téléphonique
- Adressage à plat
  - le format de l'adresse n'a aucune signification particulière quant à la localisation de l'entité communicante
  - exemple : adresse MAC



# Les techniques d'adressage

---

- Pas de champ d'adressage dans le message
  - liaison type point-à-point : seules deux entités peuvent communiquer
  - remplacé par un numéro de voie logique négocié dans une phase d'initialisation dans le mode connecté
- Adresse destination uniquement
  - l'émetteur n'a pas besoin d'être connu ou l'est déjà
  - exemple : relation maître/esclave
- Adresse source uniquement
  - messages de diffusion ou type maître/esclave
- Adresses source et destination
  - cas le plus fréquent (@src nécessaire pour les ACK si mode non connecté)



# Nommage

---

- Le nommage permet de désigner un objet
  - dissocie l'objet de sa localisation dans le réseau
  - un objet peut être déplacé sans changer de nom
- Nommage hiérarchique
  - le nommage est organisé en domaines
  - exemple : Internet (domaine .fr, .edu, .com, .gouv.fr, .education.gouv.fr, ...)
- Nommage à plat
  - il faut garantir l'unicité du nom sur l'ensemble du réseau
  - exemple : NetBios



# Nommage et annuaire

---

- Pour localiser un objet, il faut un annuaire qui fasse la correspondance nom/adresse :  
résolution de nom
- Deux méthodes
  - consultation d'un fichier local (ex. /etc/hosts)
  - consultation d'un annuaire distant sur un serveur de noms (ex. Domain Name System)



# Acheminement dans le réseau

---



# Routeur et fonction de routage (1)

---

- Routeur : nœud du réseau qui joue un rôle dans l'acheminement des **paquets** vers le destinataire final (niveau 3 OSI)
- But : trouver une route pour acheminer un paquet vers sa destination finale
  - à priori, aucun routeur n'a une vision globale de la route que prendront les paquets : les paquets sont relayés de proche en proche jusqu'au destinataire
  - l'émetteur du paquet doit connaître le premier routeur relais
  - ensuite, chaque routeur est chargé d'acheminer le paquet vers le routeur suivant



# Routeur et fonction de routage (2)

---

## ■ Fonctions d'un routeur

- envoyer le paquet entrant vers le lien de sortie approprié
- éventuellement, stocker le paquet dans une file d'attente appropriée
- éventuellement, traiter le paquet suivant la qualité de service (QoS) requise
- éventuellement, filtrage

## ■ Problèmes posés

- Comment déployer l'information de routage dans tout le réseau ?
- Existe t-il une route optimale ? Selon quel(s) critère(s) ?
- Que se passe t-il si un lien tombe en panne ?
- Changement de la topologie ? Variation de trafic ?





# Tables d'acheminement

---

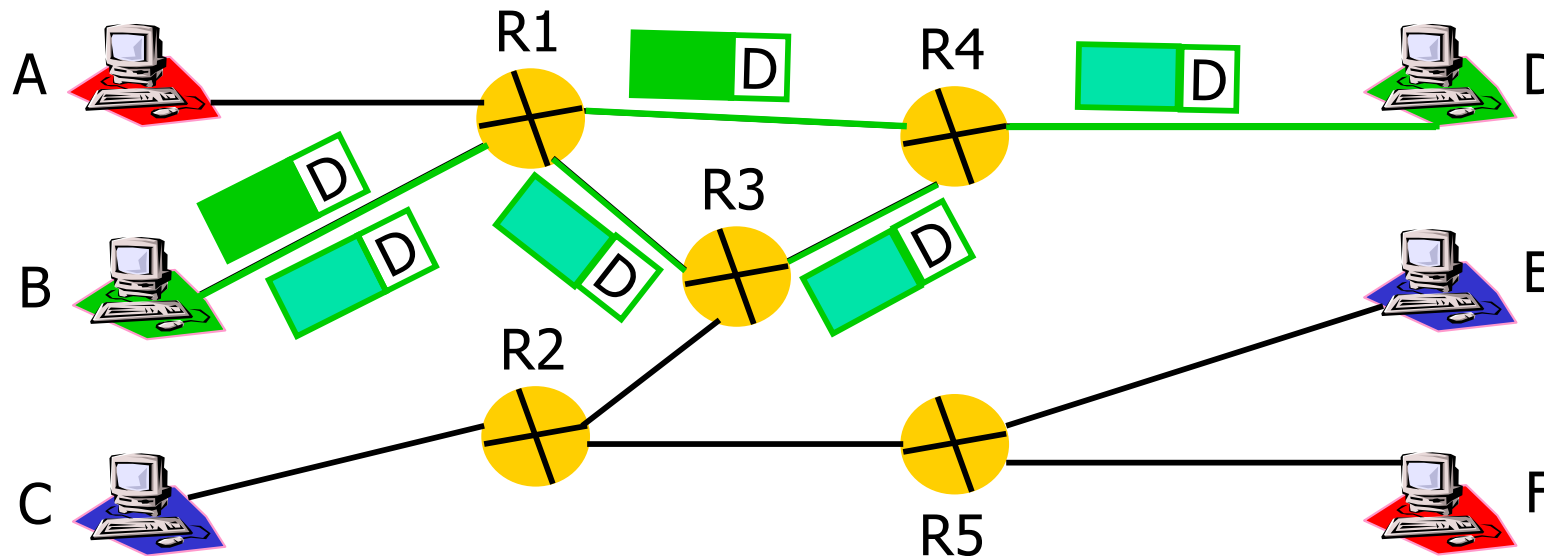
- Chaque routeur contient une table de routage ou table d'acheminement
- Chaque entrée de la table contient un triplet  
<@destination, route à prendre, coût>
  - le paquet doit contenir une information qui permet d'identifier le destinataire pour que le routeur puisse prendre une décision (@destination)
  - "route à prendre" est aussi appelé "next hop" (prochain saut)
  - "coût" permet de choisir la route appropriée si plusieurs routes sont possibles (coût pour atteindre **la destination**)

# Exemple

Comment le coût est-il déterminé ?

dest	next	coût
D	R3	3
D	R4	2
E	R3	4
F	R4	5
C	R3	3
...	...	...

dest	next	coût
D	D	1
...	...	...





# Le problème du routage

---

- Comment effectuer des choix locaux corrects dépendant d'un système global ?
  - chaque routeur doit connaître des informations sur l'état global du réseau ou du moins de ses voisins
- L'état global du réseau est
  - grand voir très grand (beaucoup d'informations)
  - dynamique (l'état du réseau dépend de la charge, de la topologie, de la défaillance de certains nœuds, ...)
  - difficile à recueillir (ne pas surcharger le réseau, périodicité du recueil d'informations ?)



# Un routage efficace c'est...

---

- une taille minimale pour les tables
  - -> consultation des tables plus rapides
  - -> consomme moins de mémoire
  - -> moins d'informations à échanger
- minimiser la fréquence des messages de contrôle générés par l'échange des informations de routage
- être robuste en évitant les boucles, la surcharge de certains routeurs...
- trouver une route optimale



# Politiques et protocoles de routage

---

- Politiques de routage : comment la route est-elle choisie ?
  - **politique déterministe** (routage statique ou non adaptatif) : la décision de routage ne dépend pas de l'état du réseau ; le choix de la route est défini une fois pour toute lors de l'initialisation
  - **politique adaptative** (routage dynamique ou adaptatif) : le chemin emprunté est fonction de données sur l'état du réseau ; les tables de routage sont régulièrement mises à jour
  - **politique mixte** : en mode orienté connexion (circuit virtuel), le chemin est établi de manière adaptative puis les données sont échangées de manière déterministe
- Protocoles de routage : comment sont construites et mises à jour les tables de routage ?



# Routage statique

---

- Routage statique
  - les tables sont mises à jour par l'administrateur uniquement quand la topologie du réseau change
  - simple mais ne recherche pas la route optimale et n'est pas adapté à la défaillance d'un lien
  - adapté aux petits réseaux et aux réseaux dans lesquels le choix de la route est limité (routes rentrées manuellement)
  - assure le séquençement des paquets même en mode non connecté (tous les paquets prennent la même route)
- Protocoles de routage statique
  - routage par le plus court chemin ou à moindre coût
  - routage par inondation



# Routage par le plus court chemin

---

- La route pour atteindre le destinataire est déterminée en calculant le plus court chemin pour atteindre le destinataire (algorithme de Dijkstra)
- Chaque lien inter-routeur a un coût (métrique) qui peut être calculé selon
  - le nombre de saut (nombre de routeurs traversés)
  - la distance réelle (en km) entre deux routeurs
  - le délai de transmission (temps de latence)
  - le nombre de paquets moyen dans les files d'attente
  - le taux d'erreurs moyen, le trafic moyen observé, ...
- Chaque routeur peut disposer de plusieurs tables de routage (une par critère) -> la route prise dépend alors de la qualité de service demandée



# Routage par inondation

---

- Chaque paquet entrant est envoyé sur tous les liens de sortie du routeur excepté celui par lequel le paquet est arrivé
  - avantages : très robuste et garantit de trouver le chemin optimal
  - inconvénients : surcharge du réseau et bouclages des paquets
    - -> numérotation des paquets par le premier routeur : un paquet avec un numéro déjà connu n'est pas réexpédié
    - -> chaque paquet contient un compteur de sauts : quand le compteur atteint zéro, le paquet est détruit
  - applications militaires
  - utilisé pour découvrir le chemin optimal et en déduire une route statique





# Routage dynamique

---

- Routage dynamique
  - les tables de routage sont mises à jour régulièrement en fonction de l'état du réseau
  - plus complexe que le routage statique et surcharge du réseau par l'échange d'informations de routage
  - permet de choisir la route optimale
  - le séquençement des paquets n'est plus assuré en mode non connecté
- Protocoles de routage dynamique
  - routage par vecteur de distance (*distance vector routing*)
  - routage à état des liens (*link state routing*)

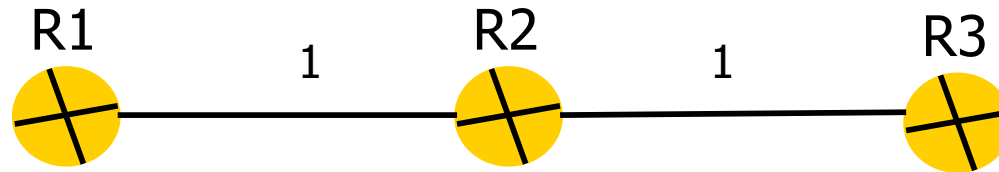


# Routage par vecteur de distance (1)

---

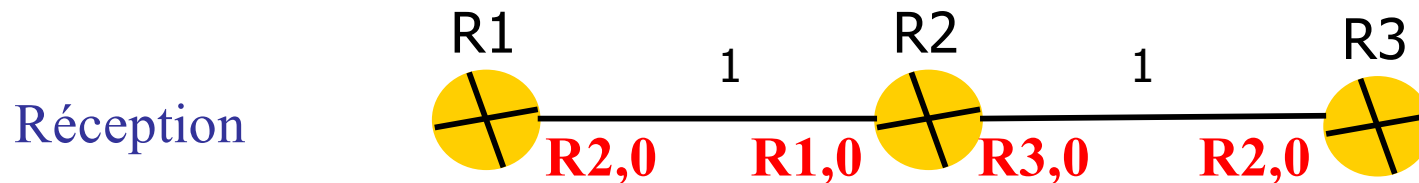
- La table de routage contient pour chaque nœud connu du réseau un coût pour joindre ce nœud
- Chaque routeur diffuse périodiquement à ses voisins sa table de routage
- Le nœud récepteur apprend ainsi qui son voisin est capable de joindre et à quel coût
- Mise à jour de la table sur le récepteur
  - Si la table reçue contient une entrée qui n'est pas dans sa table : il ajoute cette entrée dans sa table avec  $\text{coût} = \text{coût reçu} + \text{coût du lien de réception de la table}$
  - Si la table reçue contient une entrée déjà présente : il met à jour son entrée si  $\text{coût connu} > \text{coût calculé}$  ou si le coût connu à changer de valeur

# Routage par vecteur de distance (2)



Etat initial

dest	next	coût	dest	next	coût	dest	next	coût
R1	R1	0	R2	R2	0	R3	R3	0

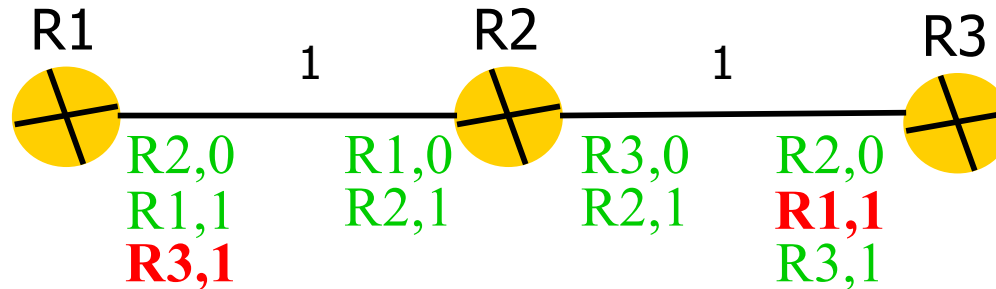


Mise à jour

dest	next	coût	dest	next	coût	dest	next	coût
R1	R1	0	R2	R2	0	R3	R3	0
R2	R2	1	R1	R1	1	R2	R2	1
			R3	R3	1			

# Routage par vecteur de distance (3)

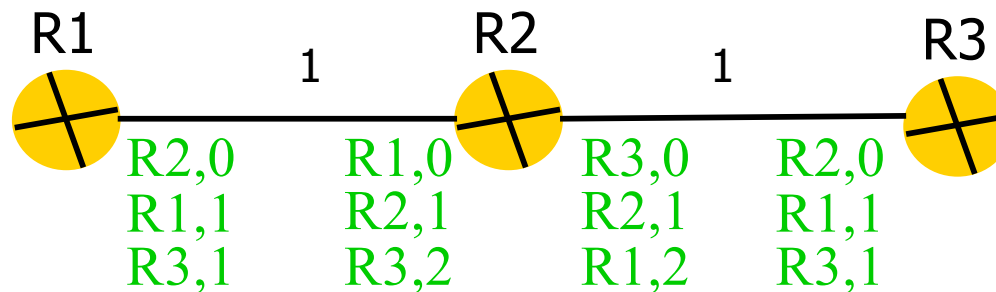
Réception



Mise à jour

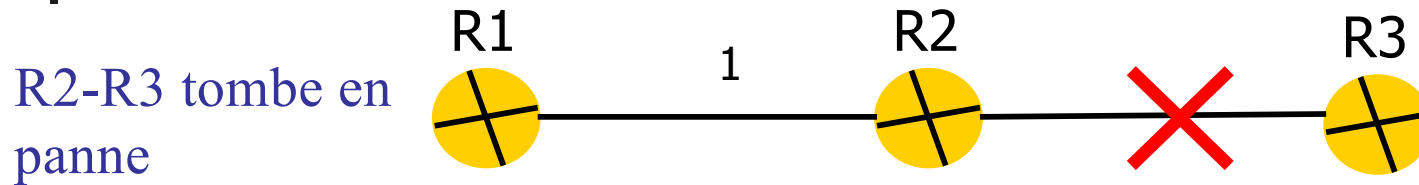
	dest	next	coût	dest	next	coût	dest	next	coût
	R1	R1	0	R2	R2	0	R3	R3	0
	R2	R2	1	R1	R1	1	R2	R2	1
	R3	R2	2	R3	R3	1	R1	R2	2

Convergence



-> aucune modification des tables

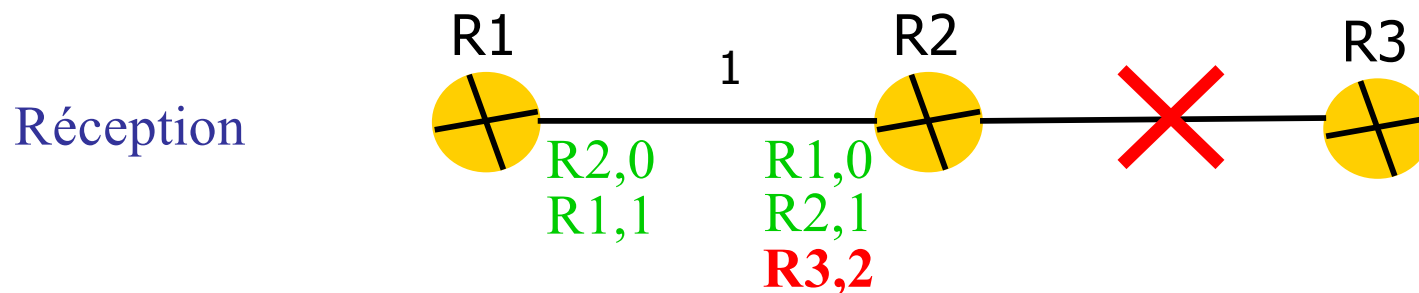
# Routage par vecteur de distance (4)



Mise à jour

	dest	next	coût	dest	next	coût
	R1	R1	0	R2	R2	0
	R2	R2	1	R1	R1	1
	R3	R2	2	R3	R3	<b>+inf</b>

R2 n'envoie plus la route vers R3 car il ne sait plus le joindre !



Mise à jour

	dest	next	coût	dest	next	coût
	R1	R1	0	R2	R2	0
	R2	R2	1	R1	R1	1
	R3	R2	2	<b>R3</b>	<b>R1</b>	<b>3</b>

R1 envoie la sienne, puis R2 envoie la sienne, ... -> comptage à l'infini



# Routage par vecteur de distance (5)

---

- Deux problèmes surviennent :
  - un bouclage apparaît dans le réseau : tous les paquets à destination de R3 oscillent entre R1 et R2
  - l'algorithme ne converge plus : à l'échange suivant, R1 apprend de R2 que désormais le coût pour joindre R3 en passant par R2 est de 3 -> il met sa table à jour (R3, R2, 4) ; de même, R2 va apprendre de R1 que désormais le coût pour joindre R3 est de 4...
- Une solution :
  - interdire à un nœud de signaler une destination qu'il connaît au routeur par lequel il l'a apprise (*split horizon*)
  - limiter la valeur infinie du coût à une petite valeur (16 dans RIP) -> convergence dès que l'infini est atteint



# Routage à état des liens (1)

---

- Problème du routage à vecteur de distance :
  - la seule info que connaît un routeur est le coût pour atteindre chaque destination
  - la convergence peut être longue sur de grands réseaux
- Principe du routage à état des liens : chaque routeur doit
  - découvrir les routeurs voisins et leur adresse réseau
  - déterminer le coût pour atteindre chaque voisin
  - construire un paquet spécial contenant son adresse, l'adresse du voisin et le coût pour l'atteindre
  - envoyer ce paquet à tous les autres routeurs
  - calculer le chemin le plus court vers chaque routeur à partir de la matrice des coûts



## Routage à état des liens (2)

---

- Chaque routeur a une vision complète de la topologie du réseau à partir d'informations distribuées
- La matrice des coûts est construite à partir des informations reçues des autres routeurs
- La topologie (graphe valué du réseau) est construite à partir de la matrice de coûts
- La table de routage est construite à partir du graphe





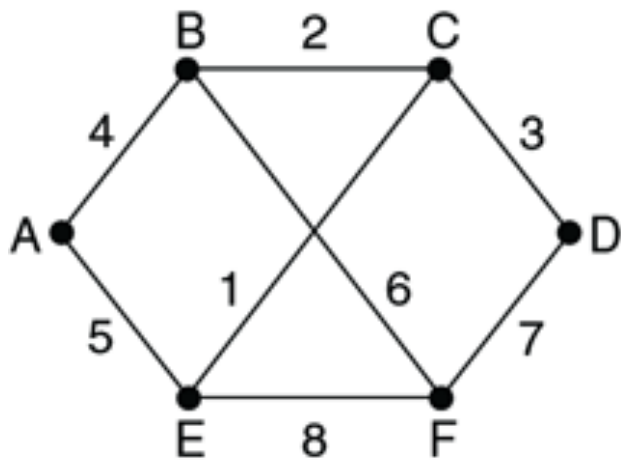
# Routage à état des liens (3)

---

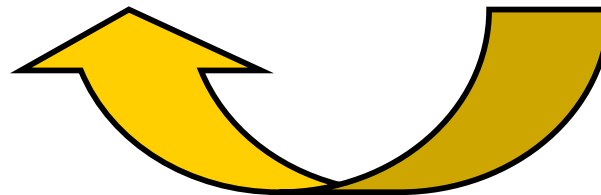
- En pratique
  - découverte des voisins : envoi d'un paquet spécial, HELLO, auquel les voisins répondent par leur identité
  - mesure du coût de la ligne : envoi d'un paquet spécial, ECHO, qui est aussitôt renvoyé par les voisins ; un timer mesure le temps A/R en tenant compte ou non de la charge du lien (temps dans les files d'attente)
  - élaboration des paquets d'état de lien : à intervalles réguliers ou quand un événement important se produit
  - distribution des paquets : par inondation avec numérotation des paquets et âge du paquet
  - calcul de la nouvelle table : algorithme de Dijkstra

# Routage à état des liens (4)

Exemple de paquets à état des liens



(a)



Lien		État		Paquets							
A		B		C		D		E		F	
Séq.		Séq.		Séq.		Séq.		Séq.		Séq.	
Âge		Âge		Âge		Âge		Âge		Âge	
B	4	A	4	B	2	C	3	A	5	B	6
E	5	C	2	D	3	F	7	C	1	D	7
		F	6	E	1			F	8	E	8

(b)

© Pearson Education France

source Tanenbaum



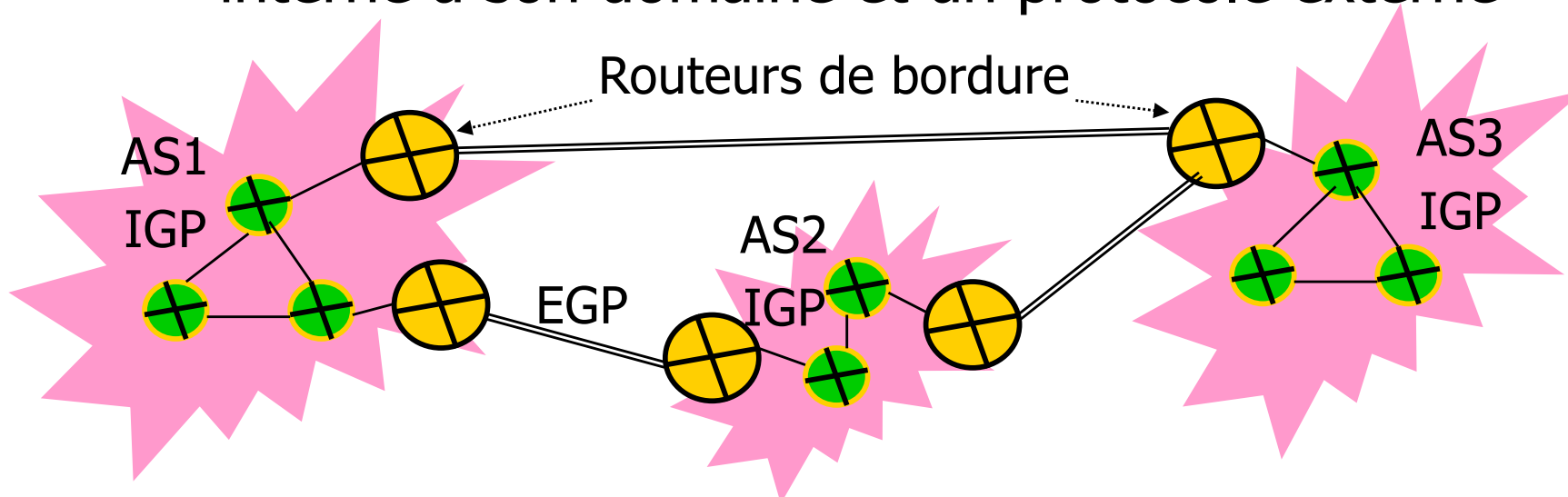
# Routage hiérarchique (1)

---

- Les protocoles de routage nécessitent la diffusion d'informations à travers le réseau
  - surcharge du réseau au détriment des données à écouler
  - plus le réseau est grand, plus les échanges sont importants, plus les tables consomment de la mémoire et plus la convergence est longue
- Routage hiérarchique
  - découpage du réseau en domaines (*AS = Autonomus System*)
  - des routeurs de bordure (*gateways* ou *edge routers*) réalisent l'interconnexion des domaines
  - limite les diffusions inhérentes au routage

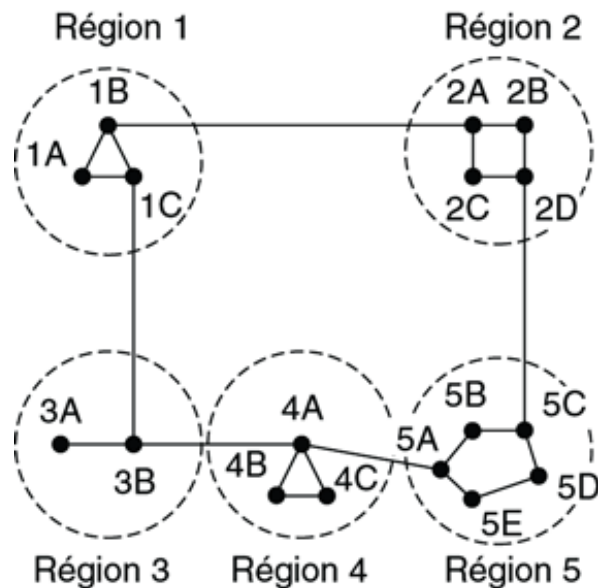
# Routage hiérarchique (2)

- Protocoles de routage intra et inter domaines
  - IGP (*Interior Gateway Protocol*) : routage interne au domaine -> ignore les nœuds externes au domaine
  - EGP (*External Gateway Protocol*) : gère l'échange d'information de routage entre domaines
  - le routeur de bordure doit supporter le protocole interne à son domaine et un protocole externe



# Routage hiérarchique (3)

Exemple (source Tanenbaum)



(a)

Table complète pour 1A

Dest.	Ligne	Sauts
1A	—	—
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Table hiérarchique pour 1A

Dest.	Ligne	Sauts
1A	—	—
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

Inconvénient lié à la perte d'info : le plus court chemin pour aller de 1A à 5C ne passe pas par 1C !



# Les principaux protocoles de routage (1)

---

- RIP (*Routing Information Protocol*)
  - protocole interne au domaine de type vecteur distance
  - largement utilisé dans Internet
  - RIP (RFC 1058), RIP-2 (RFC 1723)
- OSPF (*Open Short Path First*) - RFC 2178
  - protocole interne à état des liens
  - largement utilisé dans Internet (tend à remplacer RIP)
- IS-IS (*Intermediate System to Intermediate System*)
  - protocole de routage interne à état des liens, ISO 10589



# Les principaux protocoles de routage (2)

---

- EIGRP (*Enhanced Interior Gateway Routing Protocol*)
  - protocole propriétaire CISCO de type vecteur distance
  - utilise une métrique complexe fonction du délai d'acheminement, du débit, de la fiabilité et de la charge du réseau
- EGP (*Exterior Gateway Protocol*) - RFC 827
  - premier protocole externe utilisé dans Internet
  - largement utilisé dans Internet
- BGP (*Border Gateway Protocol*) - RFC 1771
  - protocole de routage utilisé dans Internet qui définit les échanges internes au domaine (iBGP) et externes (eBGP)



# Routage et commutation (1)

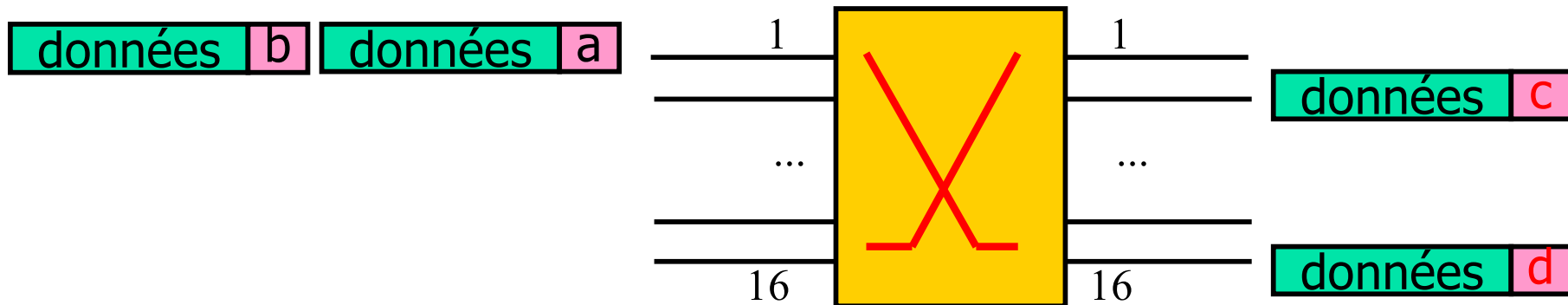
---

- Routage : je veux aller à @, quelle route dois-je prendre ?
  - en mode datagramme : la décision d'acheminement est prise pour chaque paquet et par chaque routeur traversé
  - en mode connecté : la décision d'acheminement est prise lors de l'établissement de la liaison pour le paquet d'établissement et par chaque routeur traversé
- Commutation : l'@ destination n'intervient plus dans la décision d'acheminement
  - elle est prise au niveau liaison à partir d'une autre information (adresse MAC, numéro de VLAN, numéro de circuit virtuel ou non...)
  - la décision d'acheminement est plus rapide (index dans la table de commutation)



# Routage et commutation (3)

- Circuit virtuel commuté (ex. ATM et X25)

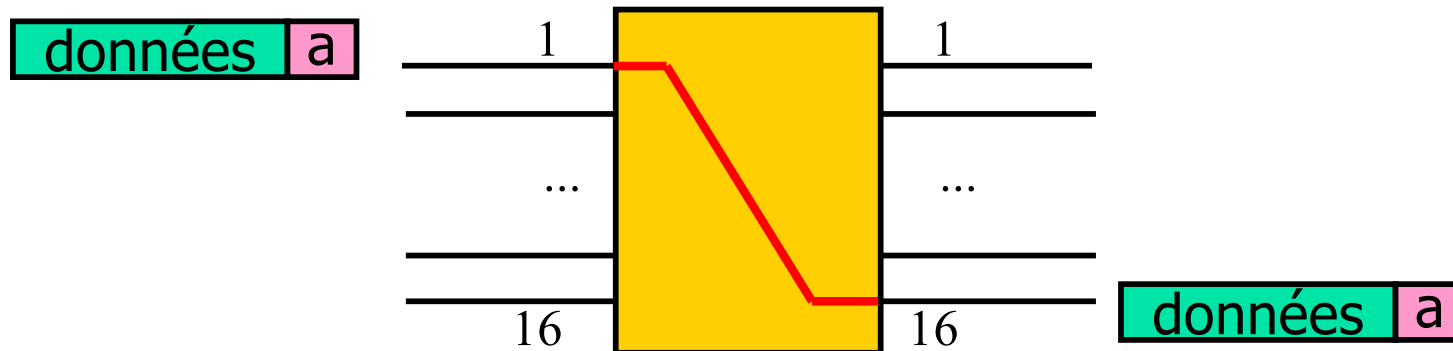


in	VC	out	VC
1	a	2	c
1	b	16	d

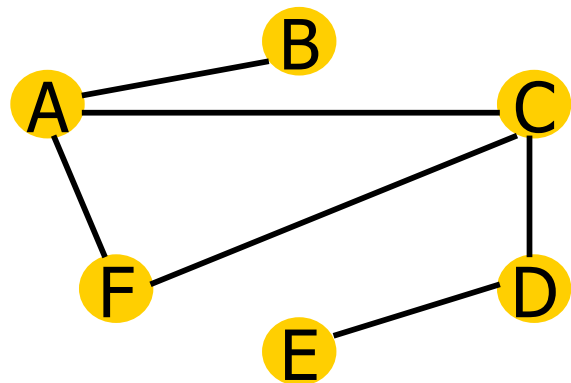
Commutation en mode connecté : la table d'acheminement est construite lors de l'établissement de la connexion

# Routage et commutation (4)

- Circuit virtuel permanent (ex. Frame Relay)



La table d'acheminement est fixée par l'opérateur



Pour aller de A à E, il faut passer par C et D



# Adaptation de la taille des unités de données





# Notion de MTU (1)

---

- MTU - *Maximum Transfer Unit*
- Taille maximale des paquets admis dans le réseau (en-tête compris)
- Cette taille varie d'un réseau à l'autre et d'une couche à l'autre
- Si un paquet a une taille supérieure à la MTU, il doit être fragmenté en plusieurs blocs à l'entrée du réseau et réassemblé à la sortie



## Notion de MTU (2)

---

- Pourquoi imposer une taille maximale ?
  - stockage des paquets dans les files d'attente des routeurs/commutateurs (mémoire limitée)
  - au matériel ou à l'OS (ex. taille des trames Ethernet)
  - au protocole (nombre de bits réservés au champ longueur du paquet)
  - limiter les retransmissions en cas d'erreurs
  - limiter le risque de congestion (temps d'occupation du canal par un paquet)
- Quelques MTU en octets
  - Ethernet (1500), Frame Relay (1600), Token Ring (4464 en 4Mbps, 17914 en 16Mbps)



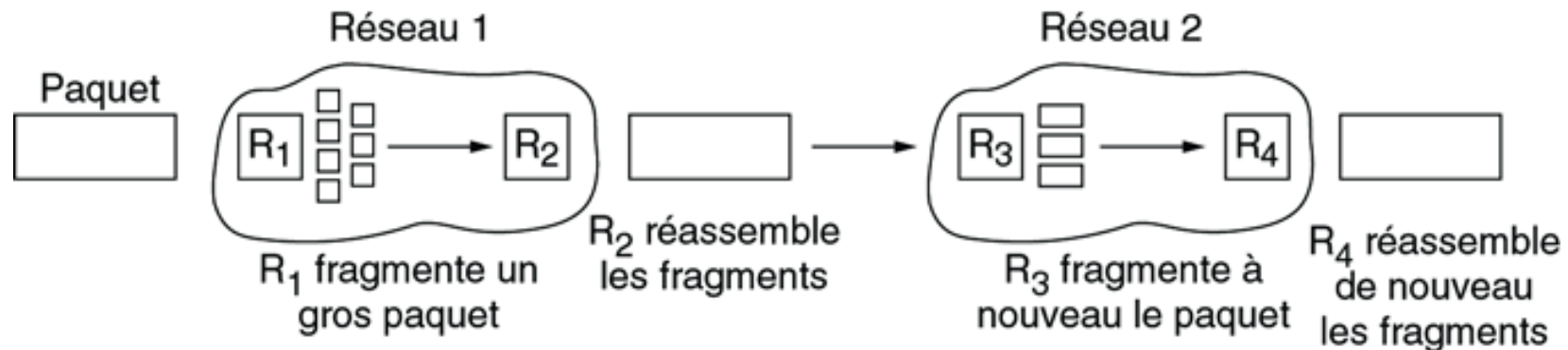
# Fragmentation et réassemblage (1)

---

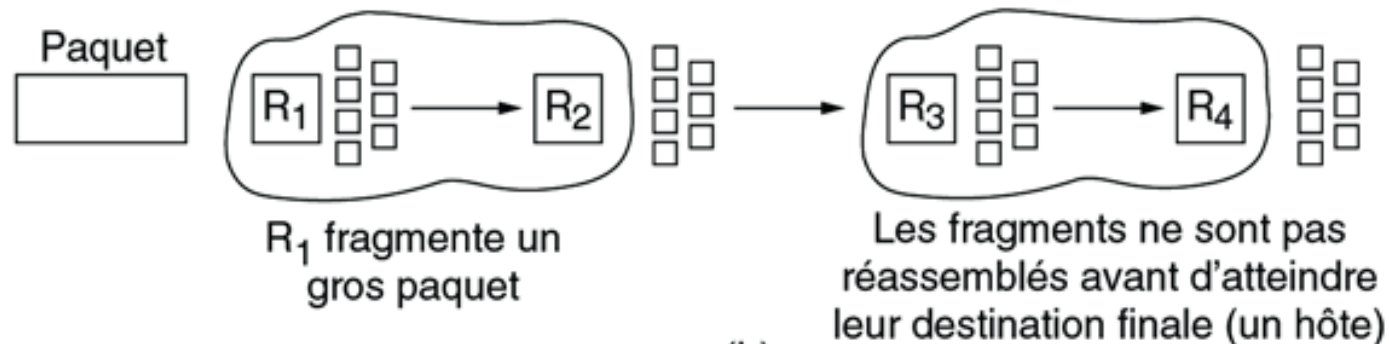
- La fragmentation est réalisée par le routeur d'entrée qui doit reconstruire un en-tête d'acheminement pour chaque fragment
- Mode non connecté
  - les différents fragments peuvent arrivés dans le désordre
  - les fragments doivent être numérotés et comptés
  - le réassemblage ne peut se faire que sur le destinataire qui doit connaître le nombre de fragments attendus
- Mode connecté
  - le routeur de sortie peut reconstituer le paquet fragmenté
  - chaque fragment contient uniquement un bit (*bit More*) d'information indiquant s'il s'agit du dernier fragment

# Fragmentation et réassemblage (2)

a) fragmentation transparente mais coûteuse



(a)



(b)

© Pearson Education France

b) fragmentation non transparente (obligatoire en mode datagramme)

# Numérotation des fragments

Numéro du premier fragment  
élémentaire dans ce paquet

Supporte les fragmentations  
multiples !

Numéro  
de paquet

Bit de fin  
de paquet

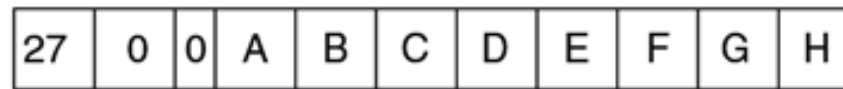
1 octet



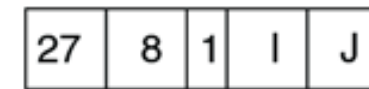
En-tête = 2 octets

En-tête

(a)

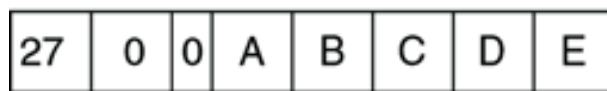


En-tête

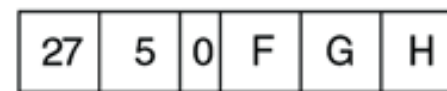


En-tête

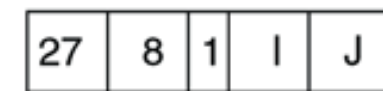
(b)



En-tête



En-tête



En-tête

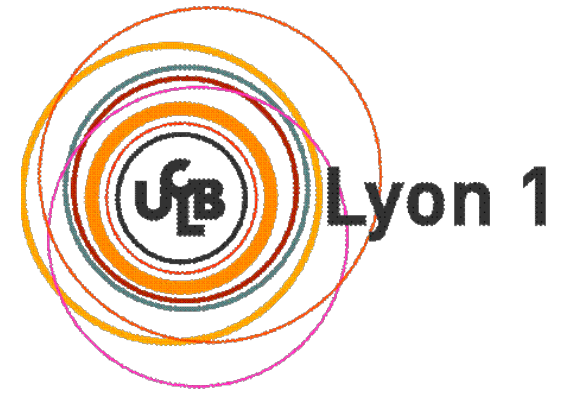
(c)

a) paquet d'origine

b) 1er réseau traversé : MTU = 10 octets de données © Pearson Education France

c) 2ième réseau traversé : MTU = 7 octets de données





# Congestion dans le réseau

---

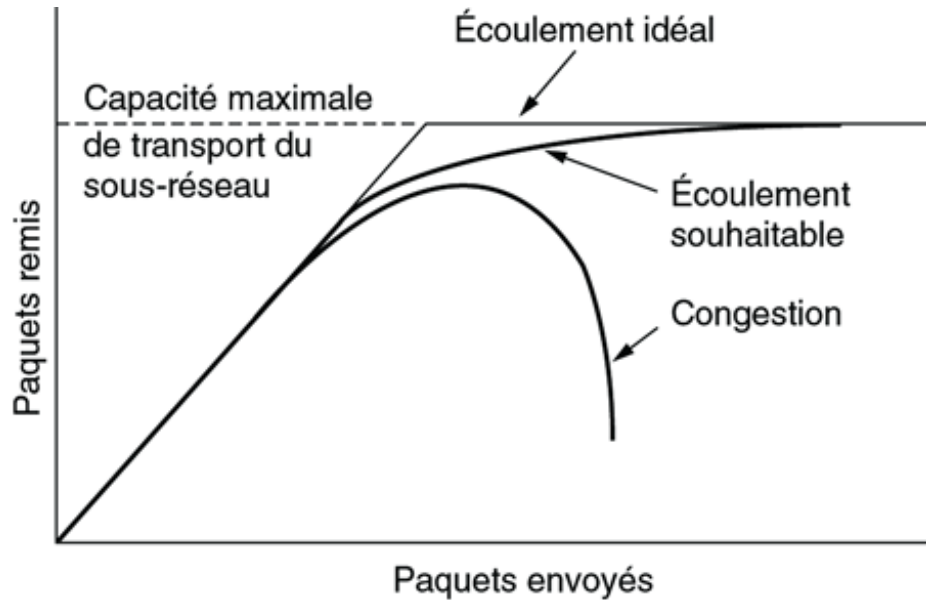


# Qu'est ce que la congestion ?

---

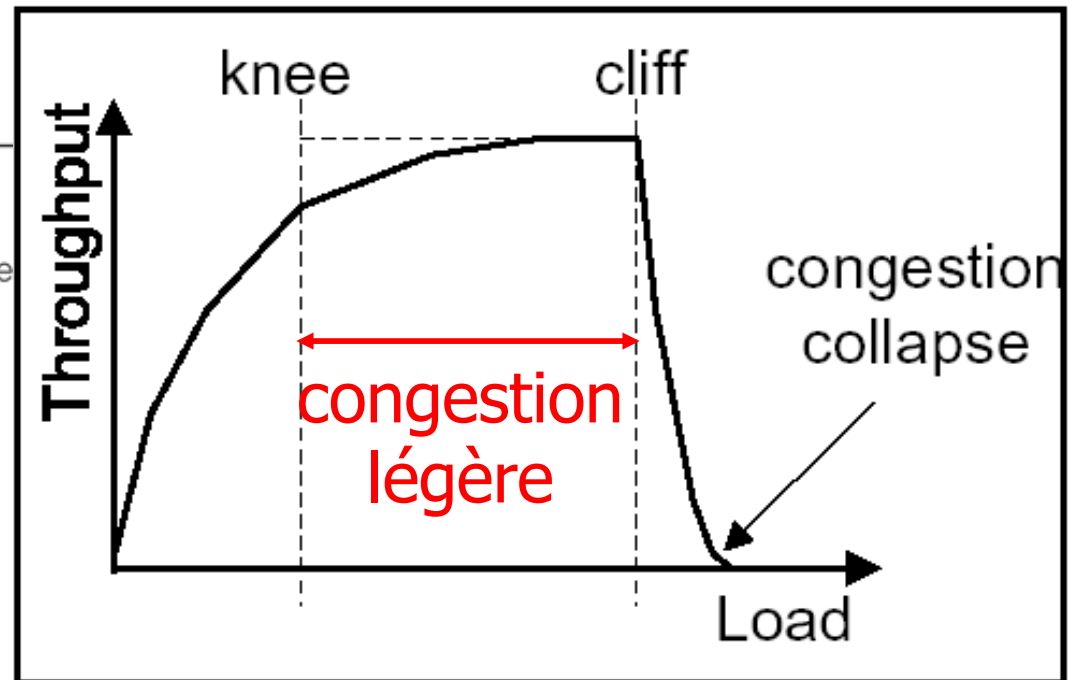
- Les routeurs/commutateurs ont des files d'attente finies !
- Congestion légère
  - le trafic augmente -> les files d'attente des nœuds intermédiaires se remplissent -> le temps d'attente avant traitement dans les nœuds augmente -> le temps de transit dans le réseau augmente -> le débit des sources diminue globalement
- Congestion sévère
  - congestion légère -> les paquets retardés ne sont pas acquittés à temps -> ils sont retransmis -> le trafic augmente d'autant plus -> les files d'attente débordent -> le réseau s'effondre, tous les paquets sont perdus !
- Mettre des files d'attente infinies n'est pas la solution !

# Congestion légère/sévère



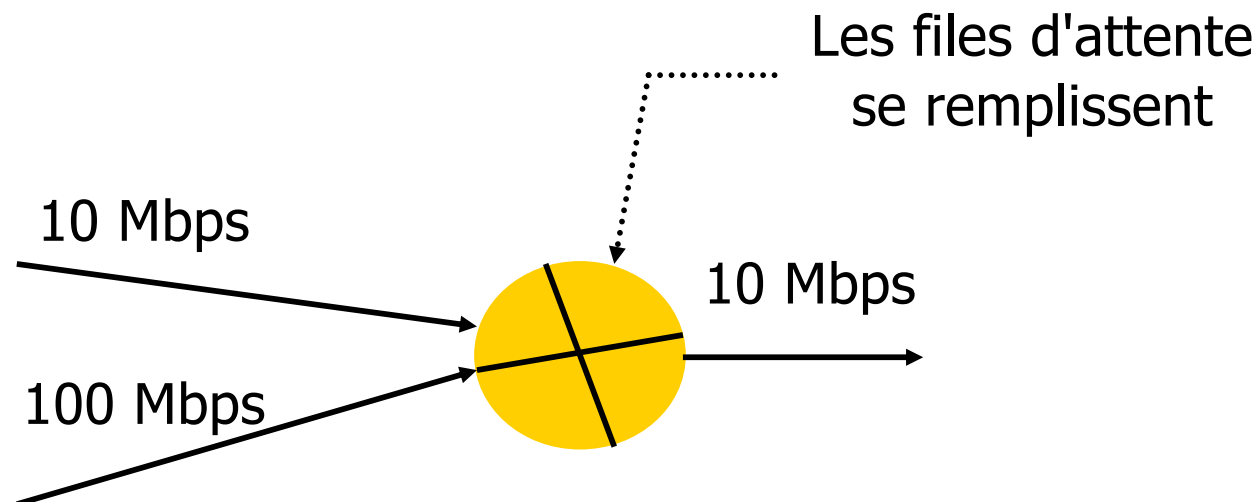
© Pearson Education France

source Pham



# Causes de congestion

- Trop de paquets sont injectés dans le réseau et prennent des routes similaires (agrégation de trafic - débit supporté par le lien)
- Différence de puissance de traitement d'un routeur à l'autre (puissance CPU)
- Emissions irrégulières des sources (rafales)





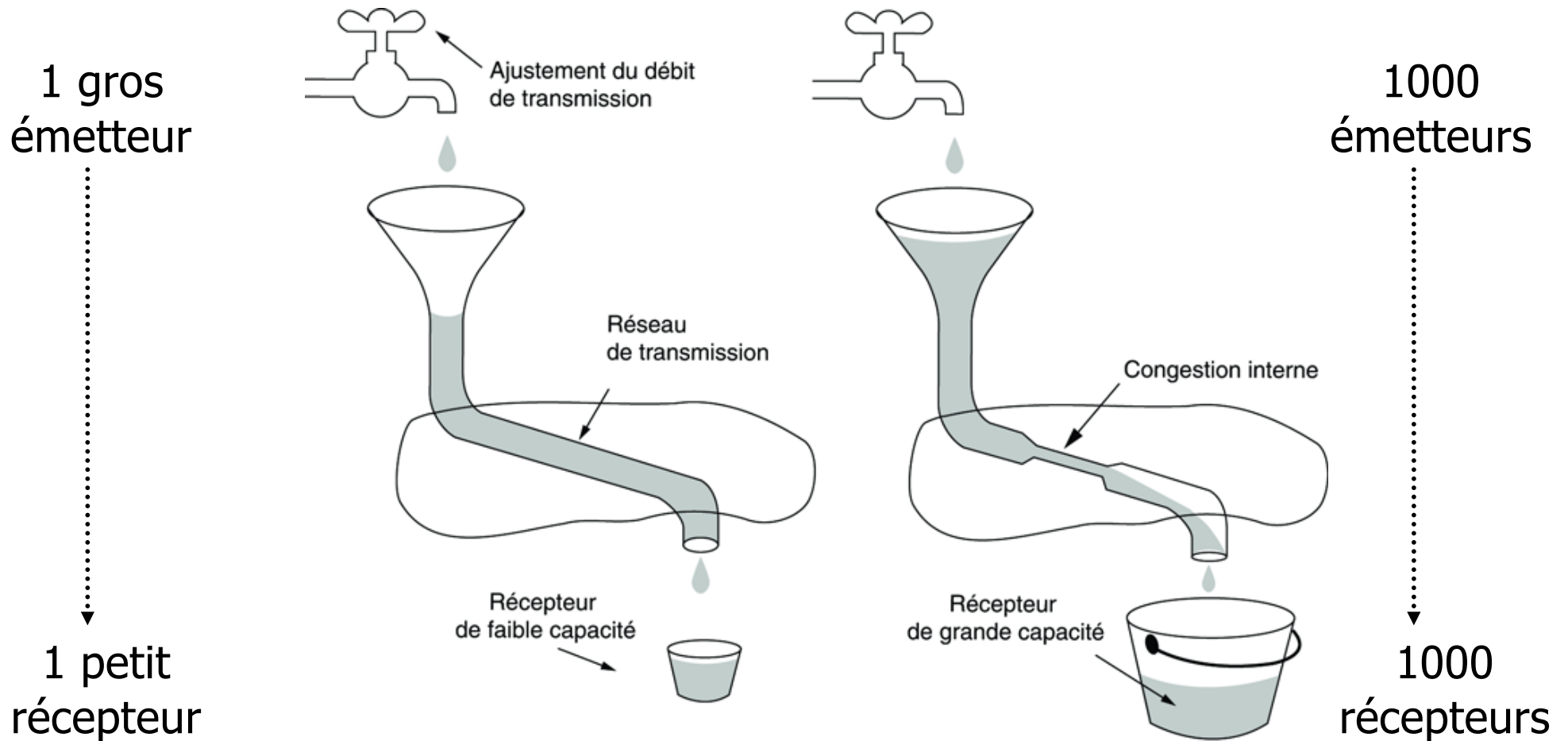
# Contrôle de flux/congestion (1)

---

- Ce sont des notions différentes !
- Contrôle de flux (rappel)
  - l'émetteur utilise une fenêtre d'anticipation qui lui permet d'envoyer plusieurs messages sans acquittement du récepteur
  - évite la surcharge du récepteur
  - concerne l'état d'une communication point-à-point
- Contrôle de congestion
  - évite que le réseau ait un trafic à écouler supérieur à ses capacités
  - cherche à limiter le nombre de paquets en transit dans le réseau
  - concerne l'état global du réseau

# Contrôle de flux/congestion (2)

- Le contrôle de flux participe à la prévention de la congestion en limitant le débit de certaines sources trop bavardes mais ne le résout pas complètement



# Contrôle de congestion

[http://wps.aw.com/aw\\_kurose\\_network\\_2/0,7240,227091-,00.html](http://wps.aw.com/aw_kurose_network_2/0,7240,227091-,00.html)



- Contrôle préventif - éviter la congestion
  - contrôle de flux : asservir le débit des sources aux capacités de traitement de chacun des nœuds
  - contrôle d'admission : ne pas admettre plus de trafic que le réseau ne peut en écouler (contrat de trafic)
  - lissage du trafic : réguler le trafic irrégulier des sources (moins de congestion avec un trafic régulier et uniforme)
- Contrôle réactif - guérir la congestion
  - éliminer le trafic en excès (tout ou une partie)
    - traiter les paquets plus prioritaires, rejeter des paquets moins prioritaires (marquage des paquets)
  - informer les nœuds en amont et en aval (par ex. bit d'alerte dans les paquets recopié dans les ack)
  - contourner les zones encombrées



# Contrôle d'admission

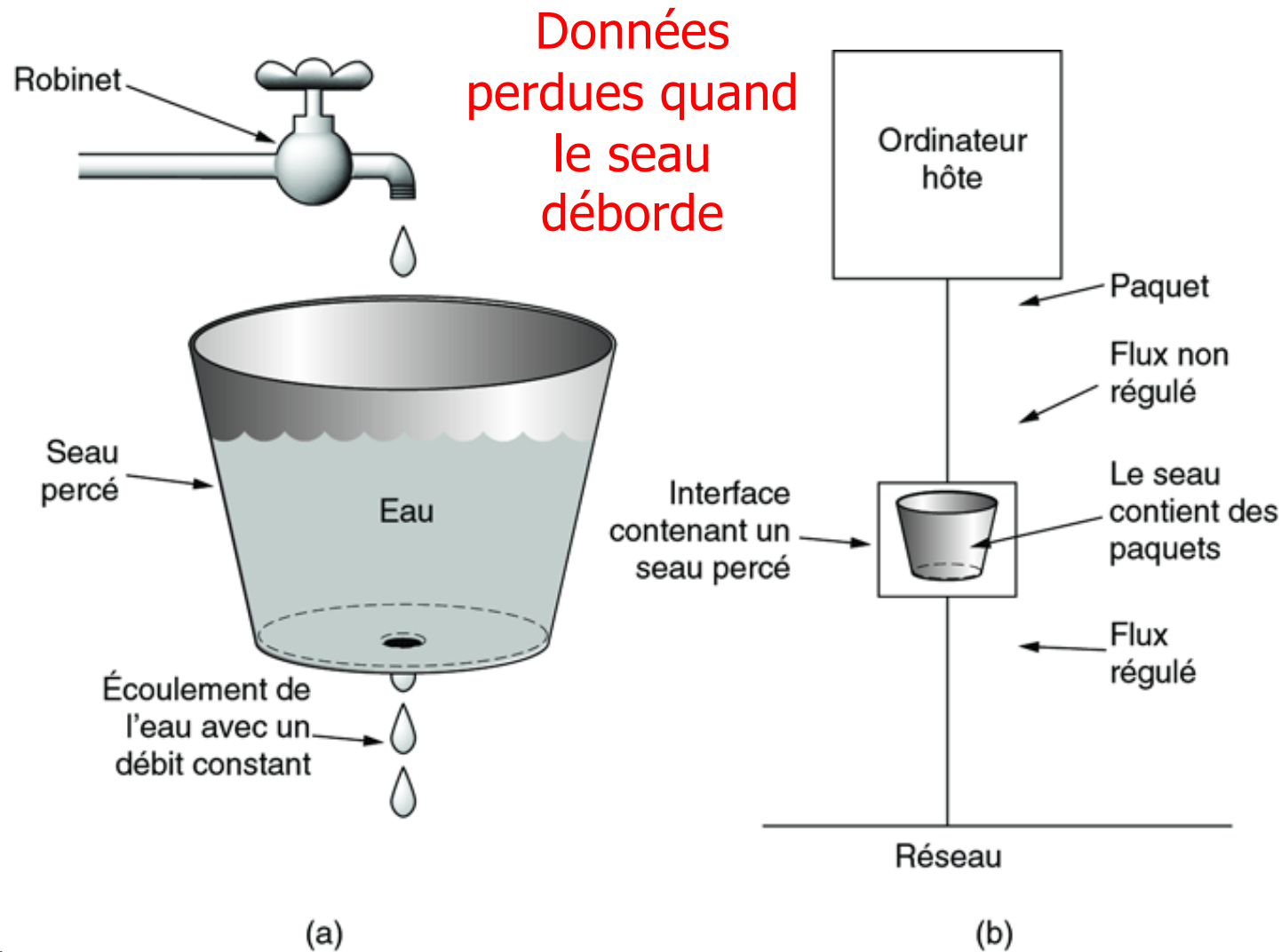
---

- En mode connecté
  - ressources allouées à chaque connexion
  - si plus de ressources, connexion refusée
  - mais problème de mutualisation des ressources !
- Autre possibilité plus souple (*Cell tagging*)
  - définir un volume moyen par source et un volume maximum toléré (lors de la connexion par exemple)
  - quand le volume moyen est dépassé les paquets sont marqués à l'entrée du réseau mais acheminés
  - au-delà du volume maximum toléré, les paquets sont éliminés à l'entrée du réseau
  - en cas de congestion, les paquets marqués sont éliminés



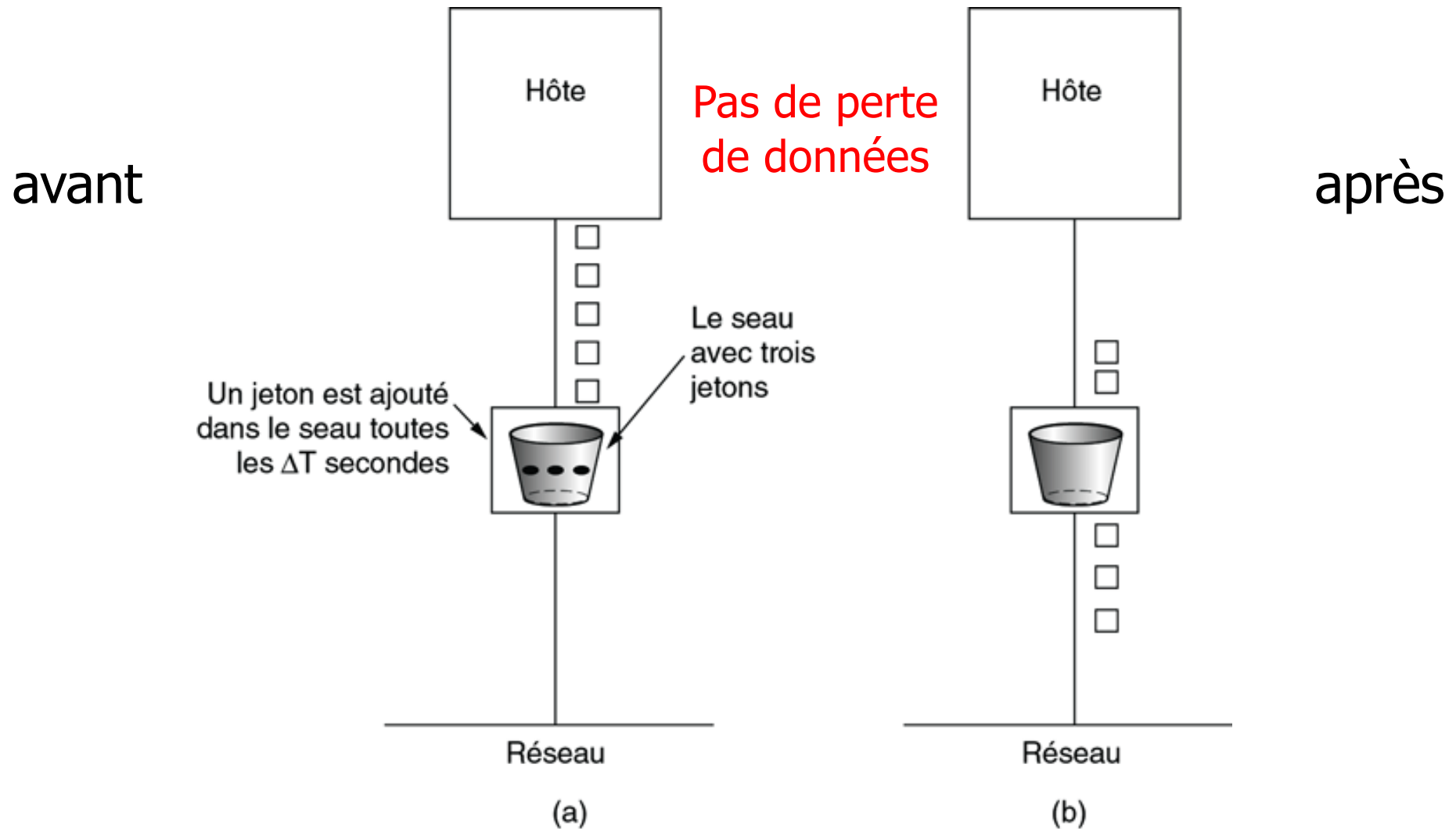
# Lissage du trafic - *Leaky Bucket*

## Algorithme du seau percé (*Leaky Bucket*)



# Lissage du trafic - *Token Bucket* (1)

## Algorithme du seau à jetons (*Token Bucket*)





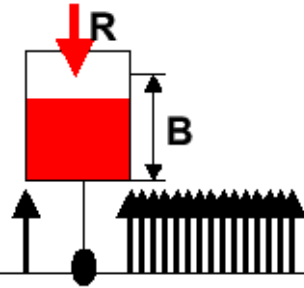
# Lissage du trafic - *Token Bucket* (2)

---

- 4 paramètres
  - capacité maximum du seau en jetons (B)
  - vitesse de génération des jetons (R)
  - capacité du lien de sortie (C)
  - taille des paquets (P)
- Exemple :  $B=4$ ,  $R=1\text{Mbps}$ ,  $P=1\text{Kb}$ ,  $C=10\text{Mbps}$ 
  - 1 jeton toutes les 1 ms
  - 1 paquet sort toutes les 0,1 ms tant qu'il y a des jetons

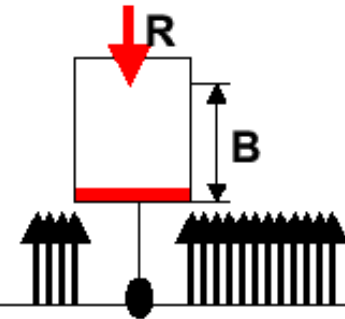
# Lissage du trafic - *Token Bucket* (3)

time = 0  
Le seau était plein de jetons avant t=0

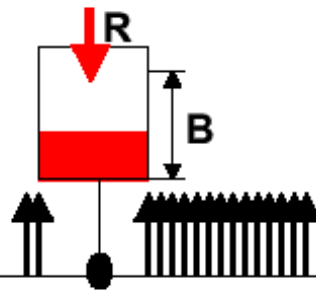


time = 0.3 ms

seau vide

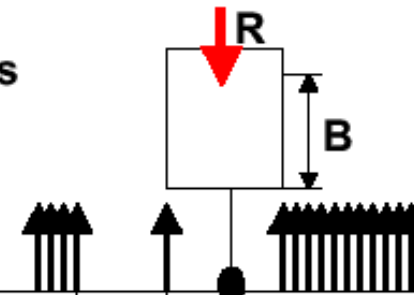


time = 0.1 ms



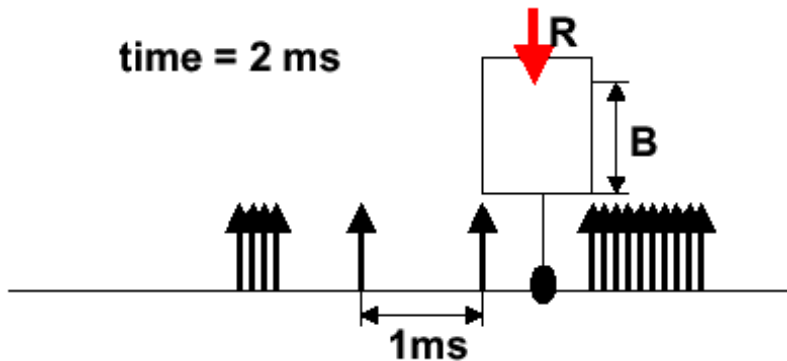
time = 1 ms

0.7ms



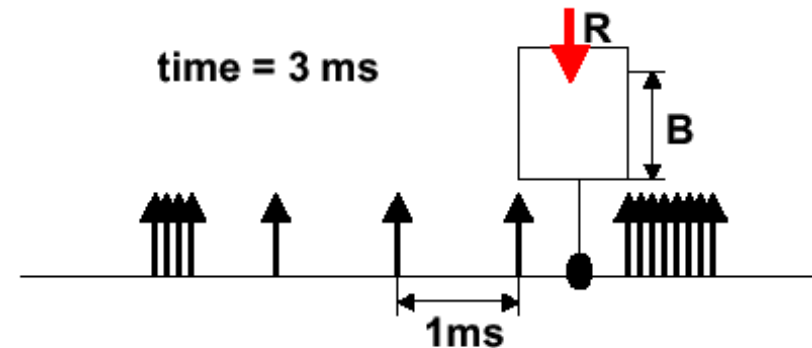
time = 2 ms

1ms



time = 3 ms

1ms





# Contrôle de congestion dans TCP (1)

---

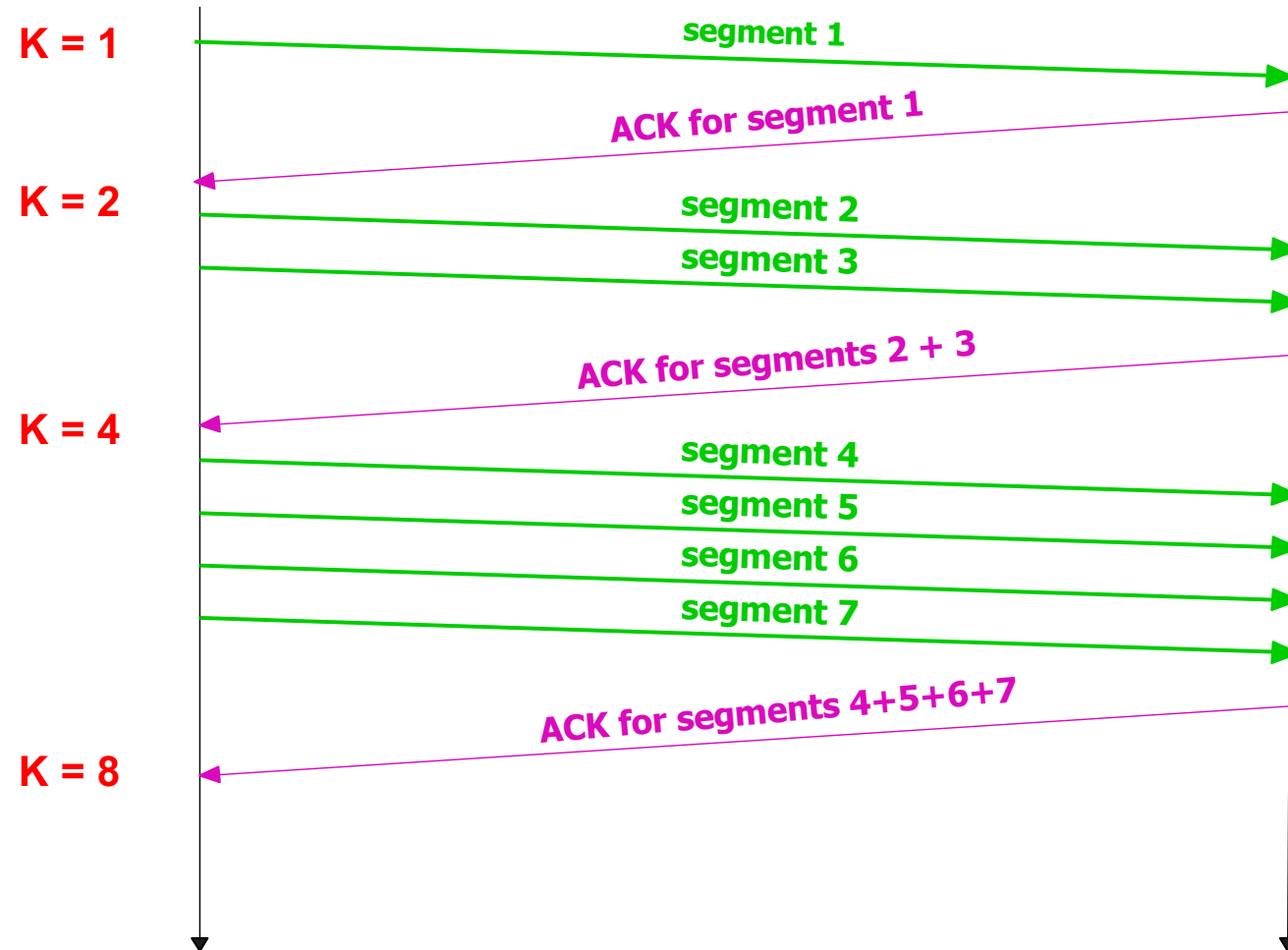
## ■ Principe

- une fenêtre de congestion sur chaque émetteur (en plus de celle de contrôle de flux par connexion)
- la quantité d'information que l'émetteur peut transmettre par anticipation est le minimum des deux fenêtres
- Soit  $K$  la taille de la fenêtre de congestion en segments
  - $K=1$  au départ ("*slow start*")
  - $K$  augmente exponentiellement jusqu'à un certain seuil  $S$  ( $S=64K$ o au départ), seuil d'évitement de congestion
  - au-delà de  $S$ ,  $K$  augmente linéairement
  - quand perte d'un paquet (*timeout*),  $K=1$  et  $S=K/2$

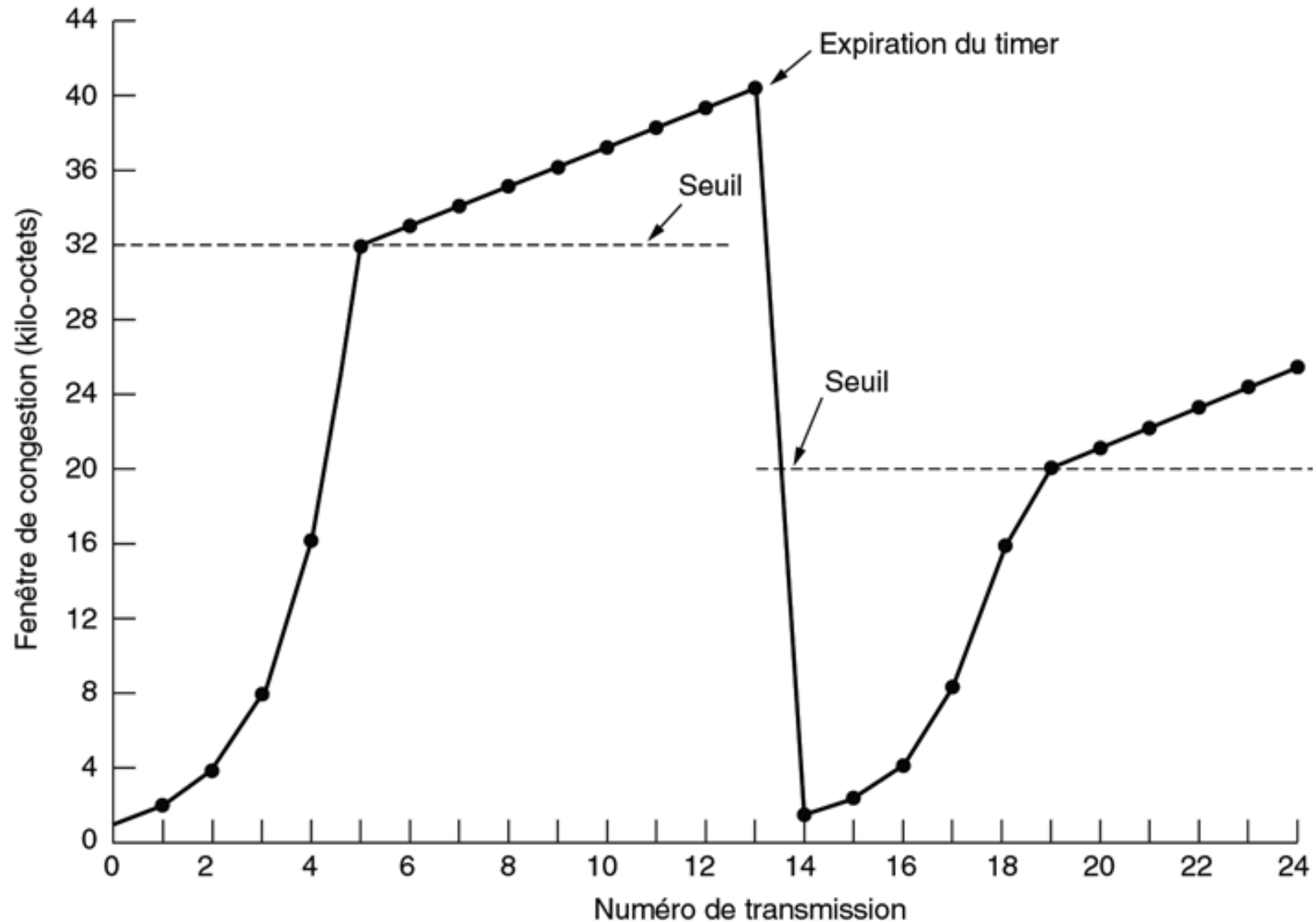
# Contrôle de congestion dans TCP (2)

## Principe du *Slow start*

Tant que tout se passe bien, la fenêtre de congestion double !



# Contrôle de congestion dans TCP (3)





# Les équipements d'interconnexion

---





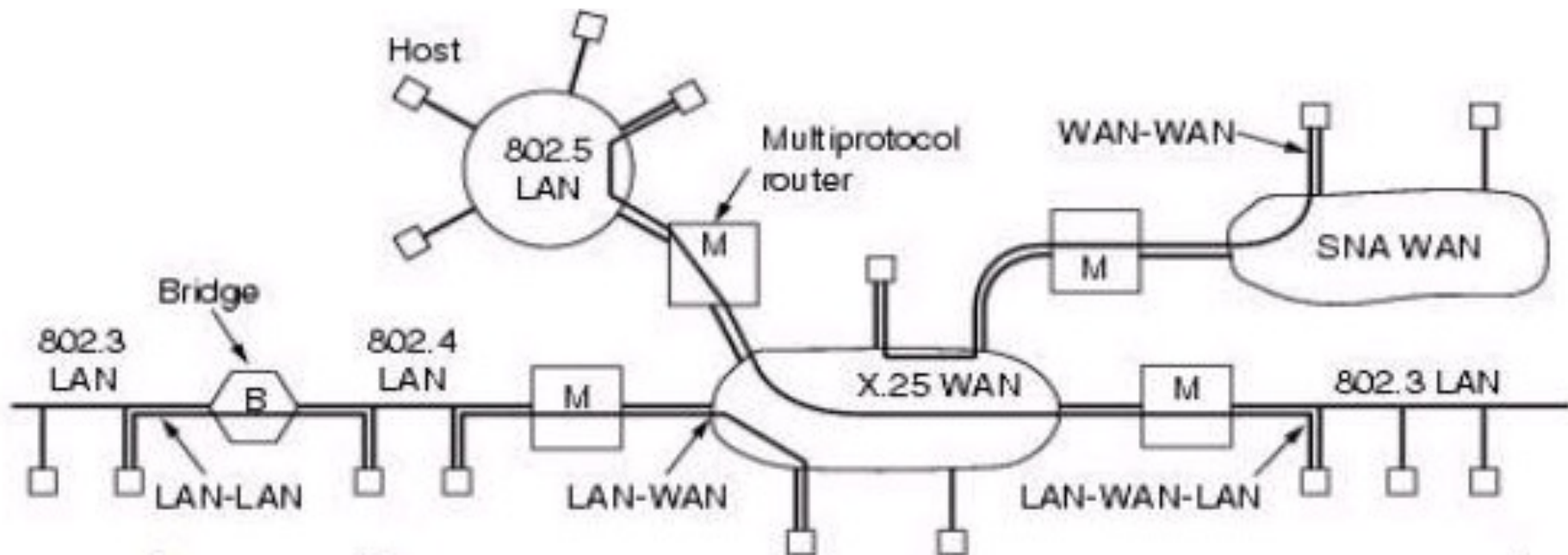
# Hétérogénéité des réseaux (1)

---

- Rien qu'au niveau de la couche réseau !
  - Service offert : avec ou sans connexion ?
  - Protocoles : IP, IPX, SNA, ATM, MPLS, Appletalk, ... ?
  - Adressage : hiérarchique (IP, ATM) ou non (MAC) ?
  - Diffusion : pas toujours supportée ?
  - Taille des paquets : MTU de chaque réseau ?
  - Qualité de Service : plein de différentes, voir absente ?
  - Gestion des erreurs : livraison fiable, ordonnée ?
  - Contrôle de flux : contrôle de débit, fenêtre glissante ?
  - Contrôle de congestion : leaky/token bucket ou autres ?
  - Sécurité : cryptage, authentification ?
  - Facturation : durée, quantité, distance... absente !
- Les équipements d'interconnexion doivent prendre en compte cette hétérogénéité !

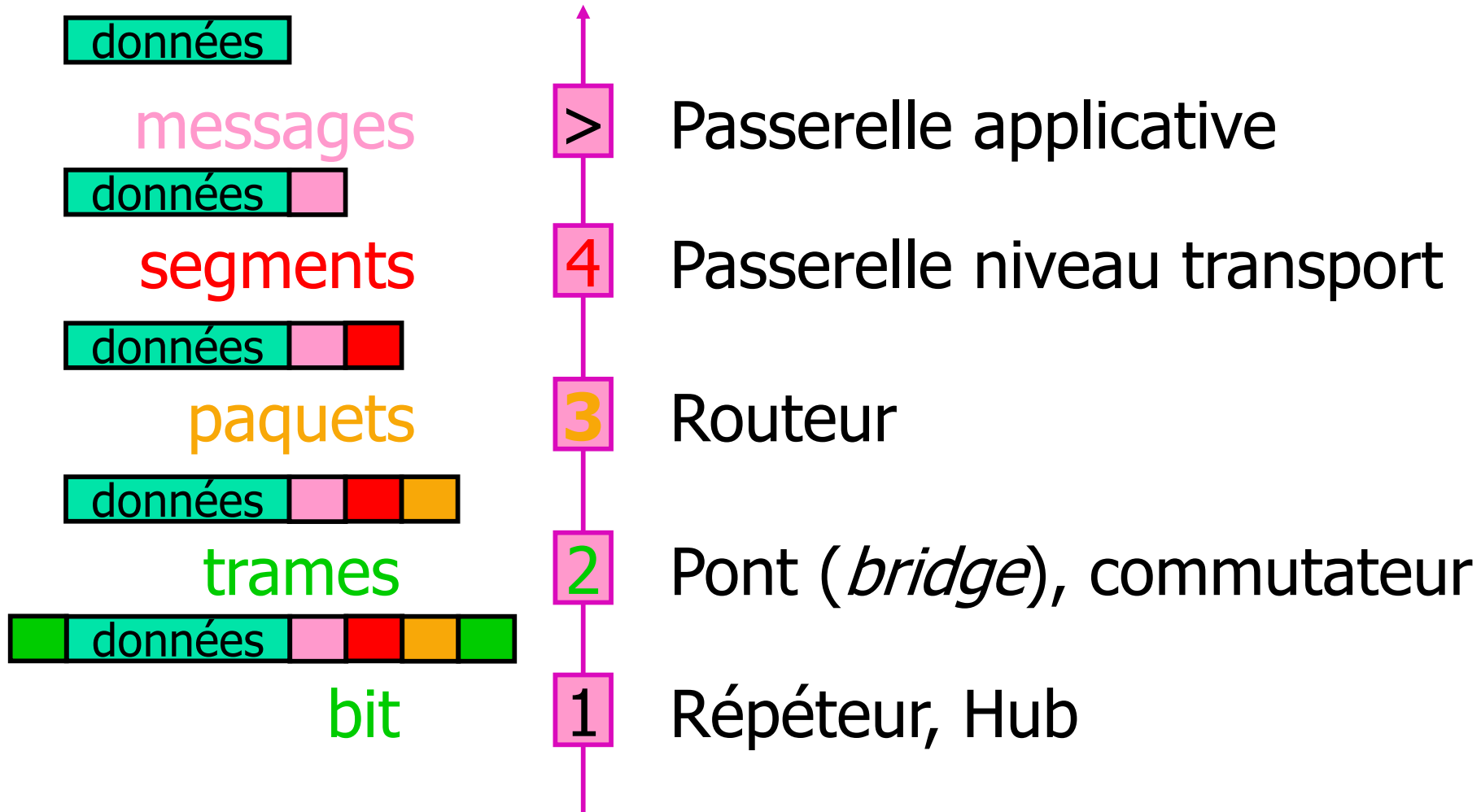
# Hétérogénéité des réseaux (2)

- Interconnexions types LAN-LAN, LAN-WAN, WAN-WAN, LAN-WAN-LAN...



# Niveaux d'interconnexion

## Niveaux OSI





# Répéteurs et Hubs

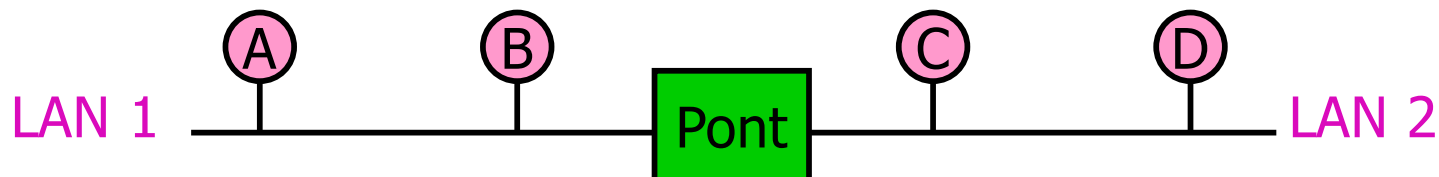
---

- Répéteur :
  - interconnexion de réseaux de même type
  - traitement au niveau bit (couche physique) entre deux segments Ethernet par ex.
  - régénération électrique du signal
  - but : augmenter la distance maximale entre deux stations en reliant deux segments, adapter deux supports différents (coaxial<-> fibre optique par ex.)
  - pas d'administration mais ne diminue pas la charge et ne filtre pas les collisions
- Hub ou concentrateur : multi-répéteurs
  - un répéteur sur chaque port
  - toutes les trames sont répétées sur tous les ports

# Ponts - rôle

## ■ Interconnexion de niveau 2

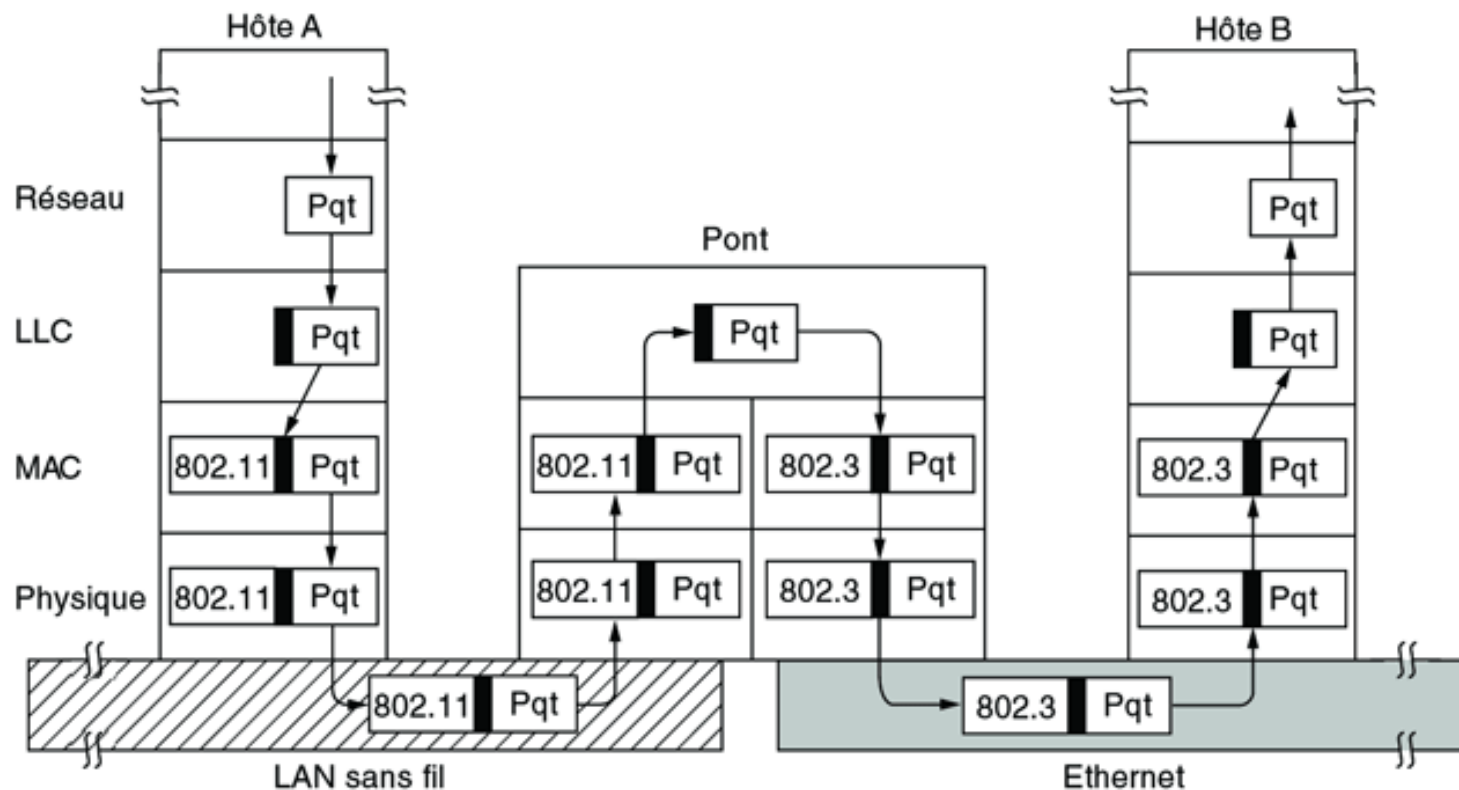
- pas d'analyse des paquets -> acheminement de tout type de paquets (IPv4, IPv6, ATM, ...)
- permet d'interconnecter différents types de LAN 802 (Ethernet, Token ring, 802.11, ...)
- permet de répartir la charge, un domaine de collision par ligne sur Ethernet
- possibilité de filtrage en fonction de l'adresse MAC
- contrôle du FCS



**Une trame A<->B (resp. C<->D) ne va pas sur LAN2 (resp. LAN1)**

# Ponts - fonctionnement

- Un pont possède autant de lignes (ou interfaces) que de LAN interconnectés ; chaque ligne contient la sous-couche MAC appropriée





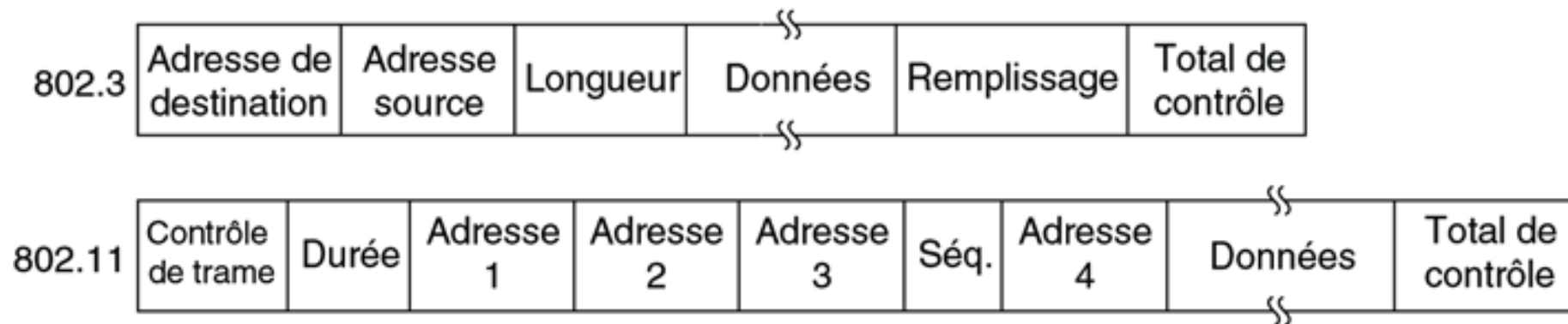
# Ponts - fonctionnement

---

- Une table de localisation basée sur l'adresse MAC permet de connaître le LAN de sortie (*FDB - Forwarding Data Base*)
- Un timer est associé à chaque entrée de la table (les stations peu bavardes sont éliminées)
- Quand un pont reçoit une trame
  - LAN destination = LAN source -> rejet de la trame
  - LAN destination  $\neq$  LAN source -> acheminement
  - LAN destination inconnue, diffusion de la trame sur toutes les lignes sauf celle d'entrée

# Ponts - reformatage des trames

- Le format de l'en-tête des trames diffère d'un LAN à l'autre
- Le pont doit reformater les trames
  - -> consomme du temps CPU
  - -> il faut recalculer le CRC







# Ponts - différents types

---

- Ponts simples

- table d'acheminement (FDB) statique (configurée par l'administrateur)

- Ponts transparents

- FDB construite dynamiquement et maintenue à jour par analyse des trames entrantes
- déplacements de stations transparents

- Ponts à routage par la source

- la route à suivre est indiquée par la trame elle-même
- une trame de découverte est envoyée vers le destinataire avant l'envoi des données
- champ RI dans Token Ring

- Ponts distants



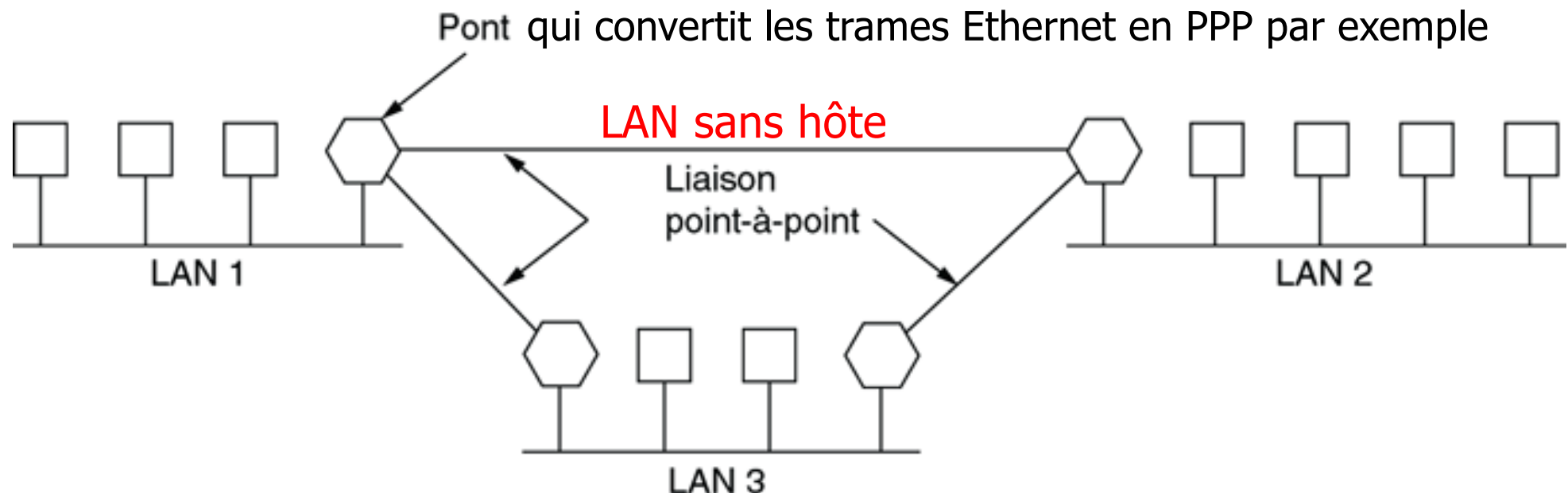
# Ponts - difficultés

---

- problème de stockage des trames
  - trames d'un LAN Gigabit Ethernet vers un LAN sans fil à 11Mbps
  - 3 LAN émettent simultanément vers un même LAN
- problème de fragmentation
  - le LAN destination a une MTU inférieure au LAN source
  - les protocoles de liaison ne font pas de fragmentation donc rejet des trames trop longues
- problème de sécurité et qualité de service
  - cryptage des données au niveau 2 dans 802.11 mais pas dans Ethernet
  - idem pour la qualité de service

# Ponts distants

- Interconnexion de LAN distants de plusieurs centaines de kilomètres par des liaisons point à point (liaisons louées par exemple)



© Pearson Education France

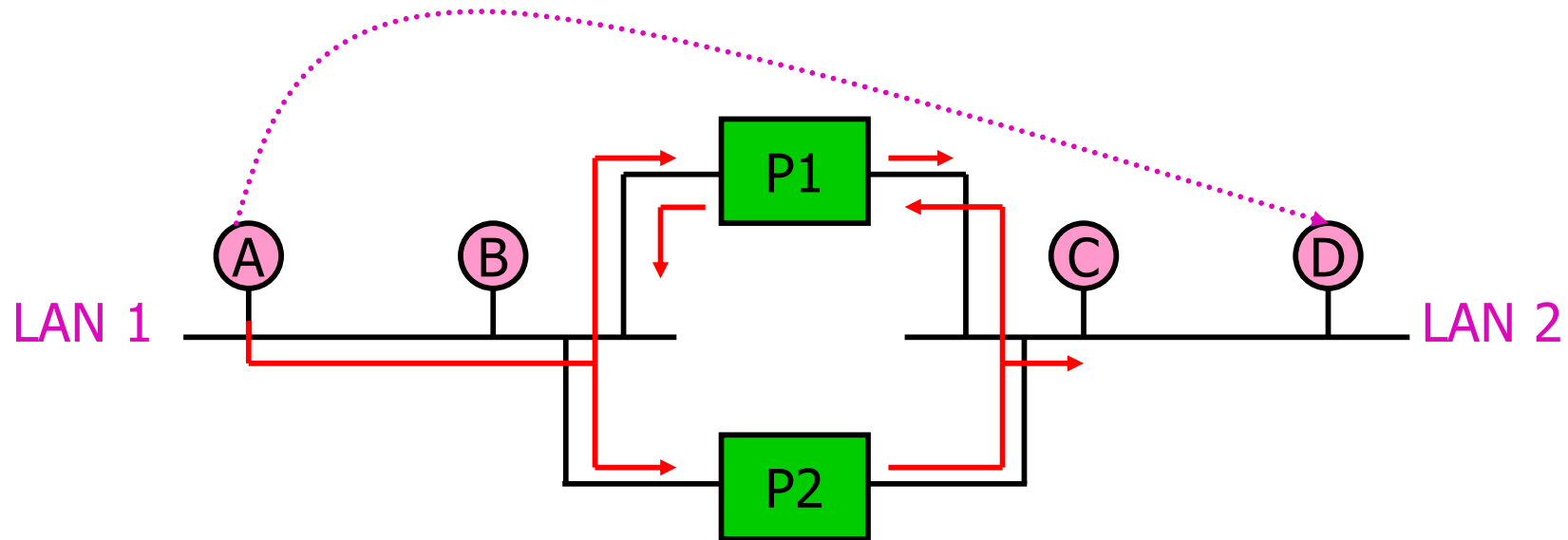


# Ponts/commutateurs

---

- Pas vraiment de différence !
- Dans les deux cas, sur Ethernet, une ligne = un domaine de collisions
- On parle plutôt de commutateur quand on interconnecte des machines individuelles et de pont pour l'interconnexion de LAN
- Un commutateur possède en général plus de lignes qu'un pont

# Ponts redondants (1)



A envoie une trame à D qui n'est pas encore localisée par P1 et P2 (D n'a pas encore émis de trame)

- P1 et P2 diffusent la trame sur le LAN2
- P1 capte la trame diffusée par P2 et la rediffuse sur le LAN1 (de même P2 rediffuse celle de P1)
- et ainsi de suite...



## Ponts redondants (2)

---

- La redondance permet une tolérance aux pannes et une meilleure efficacité mais introduit un problème de bouclage
  - duplication des trames
  - oscillations des trames de destination inconnue du fait de l'inondation
- Il faut mettre en œuvre dans les ponts un protocole qui permet de trouver un arbre couvrant (*spanning tree*)
  - arbre couvrant = mettre en place une topologie logique sans boucle à partir d'un nœud racine
  - nécessite un échange d'information entre ponts



# Routeurs (1)

---

- Permet d'interconnecter différents sous-réseaux
- Beaucoup d'intelligence (CPU, mémoire, ...)
- Rôle premier : acheminement des paquets vers le bon lien de sortie en fonction de l'adresse destination (niveau 3)
- Rôles secondaires : sécurité, qualité de service
- Une adresse par interface (connue des stations connectées sur le même sous-réseau)
- Matériel dédié (Cisco, ...) ou simple station avec plusieurs interfaces (au moins 2)



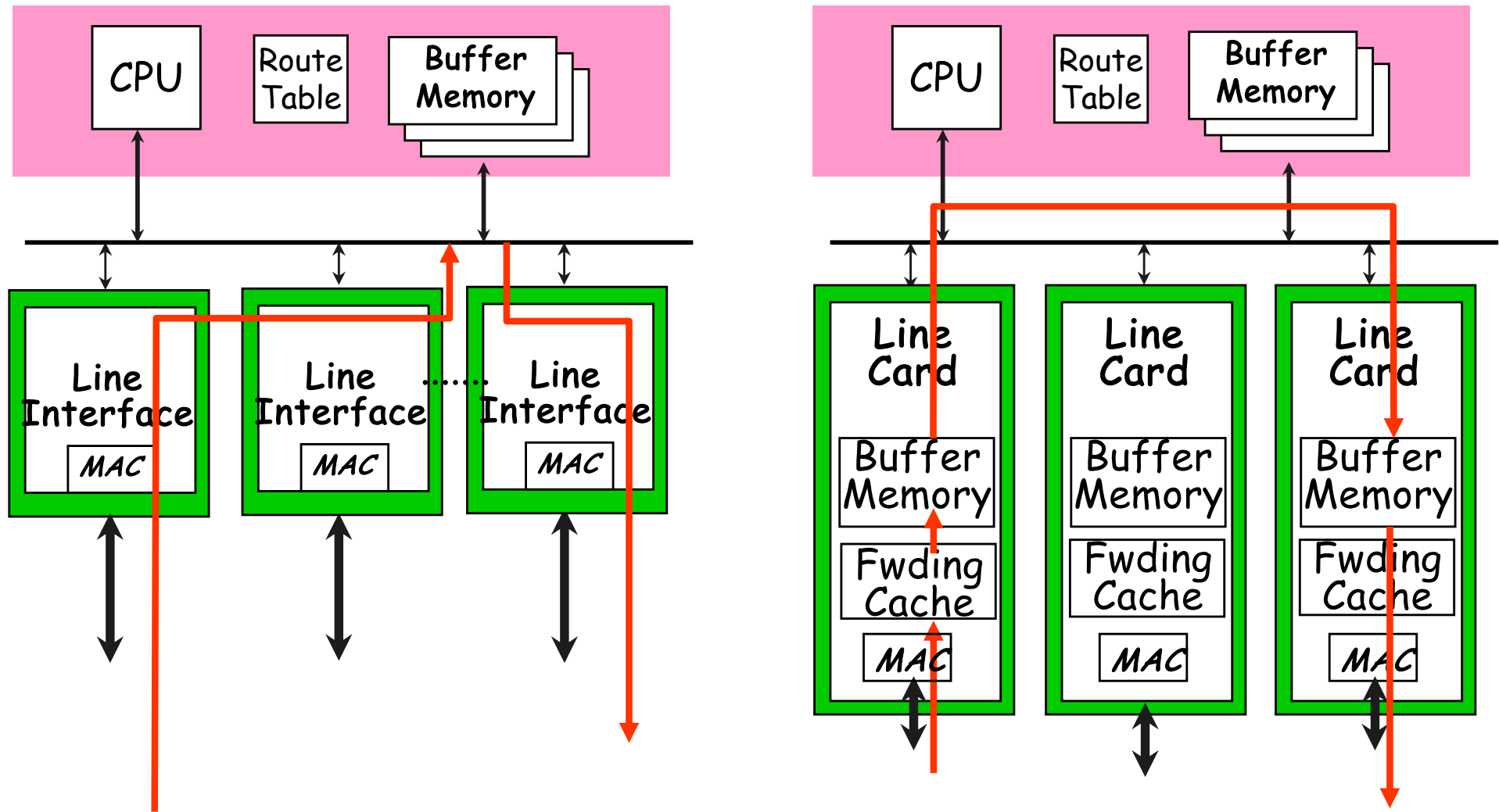
# Routeurs (2)

---

- Avantages :
  - très bon filtre
    - trames de collisions, broadcast, multicast
    - possibilité de filtrer des trames en fonction des adresses source/destination (*firewall*)
  - un routeur peut mettre en œuvre de la segmentation, du contrôle de congestion, ... (niveau 3)
- Désavantages : coût, configuration
- Routeurs multi-protocoles
  - capable de router plusieurs protocoles de niveau 3
  - conversion du format des paquets entre deux sous-réseaux de natures différentes



# Routeurs - architectures (3)



source Pham



# Ponts et routeurs

---

- Performances
  - pont plus performant mais laisse passer les messages de diffusion... -> interconnexion locale
- Sécurité
  - pont -> filtrage par adresses MAC difficile (il faut connaître toutes les adresses MAC...)
- Facilité d'administration
  - pont transparent
    - inconnu des autres stations
    - s'auto-configure
- Les B-routeurs (*Bridge-router*)
  - fonctionnalités des ponts et routeurs dans un même chassis : si protocole routable (IP), table de routage sinon (Netbios) table d'acheminement



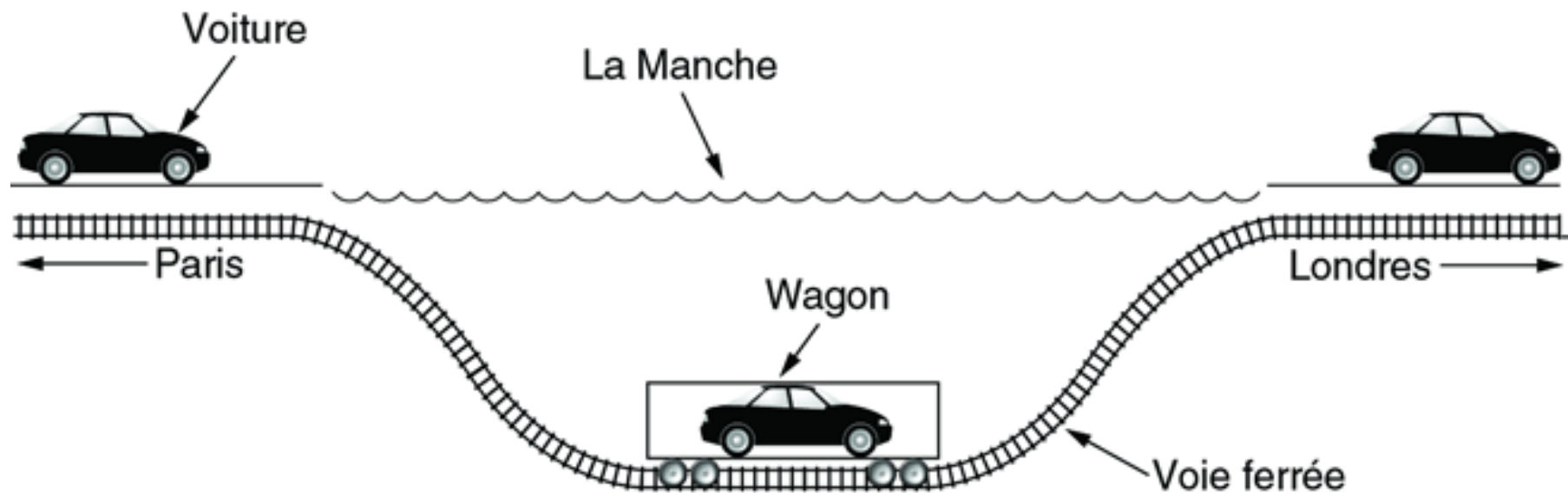
# Passerelles

---

- Tout élément d'interconnexion de niveau supérieur à la couche réseau
- Exemple de passerelle niveau transport : lien entre une connexion TCP et une connexion SNA
- Exemple de passerelle applicative : courrier électronique -> SMS

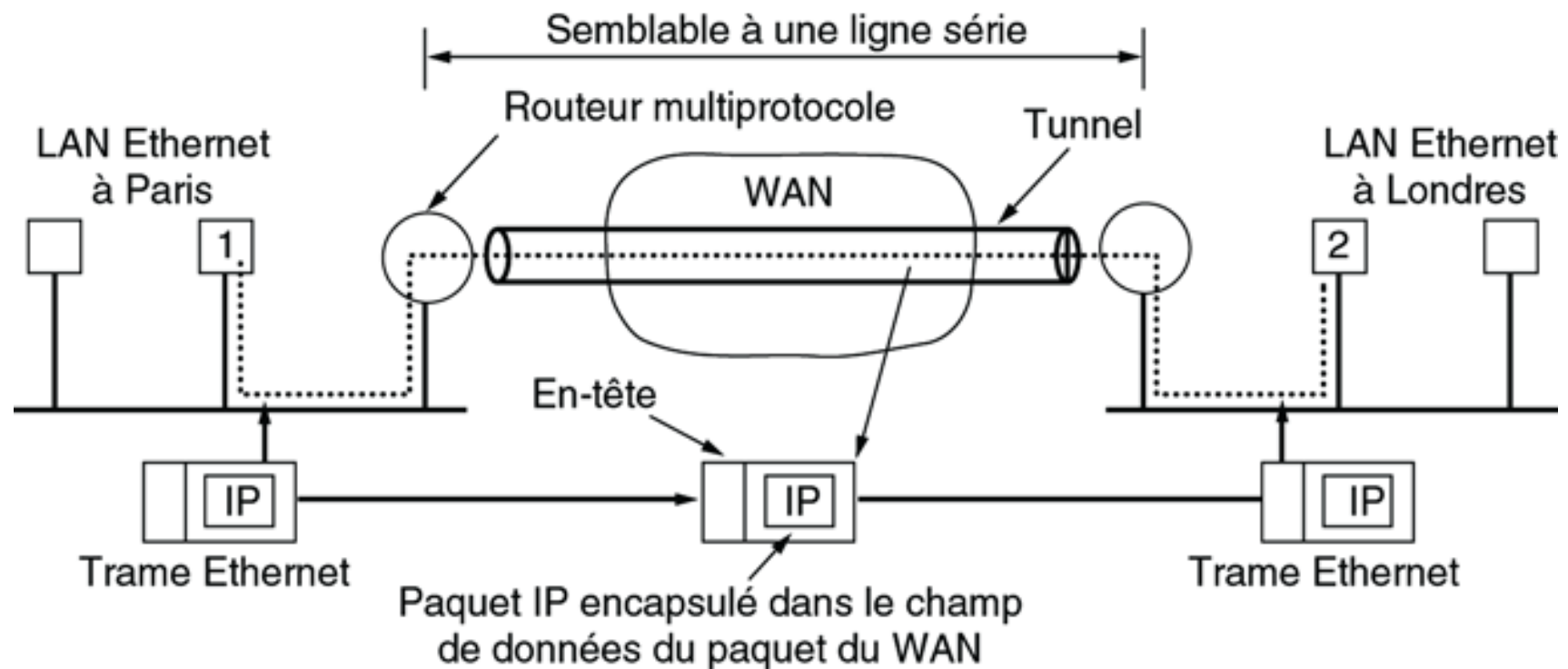
# Tunnel

- Tunnel : liaison point à point virtuelle pour transporter un protocole dans un autre
  - source et destinataire utilisant la même technologie mais séparés par un réseau d'un type différent
  - encapsulation entre deux routeurs multiprotocoles



# Tunnel - Exemple

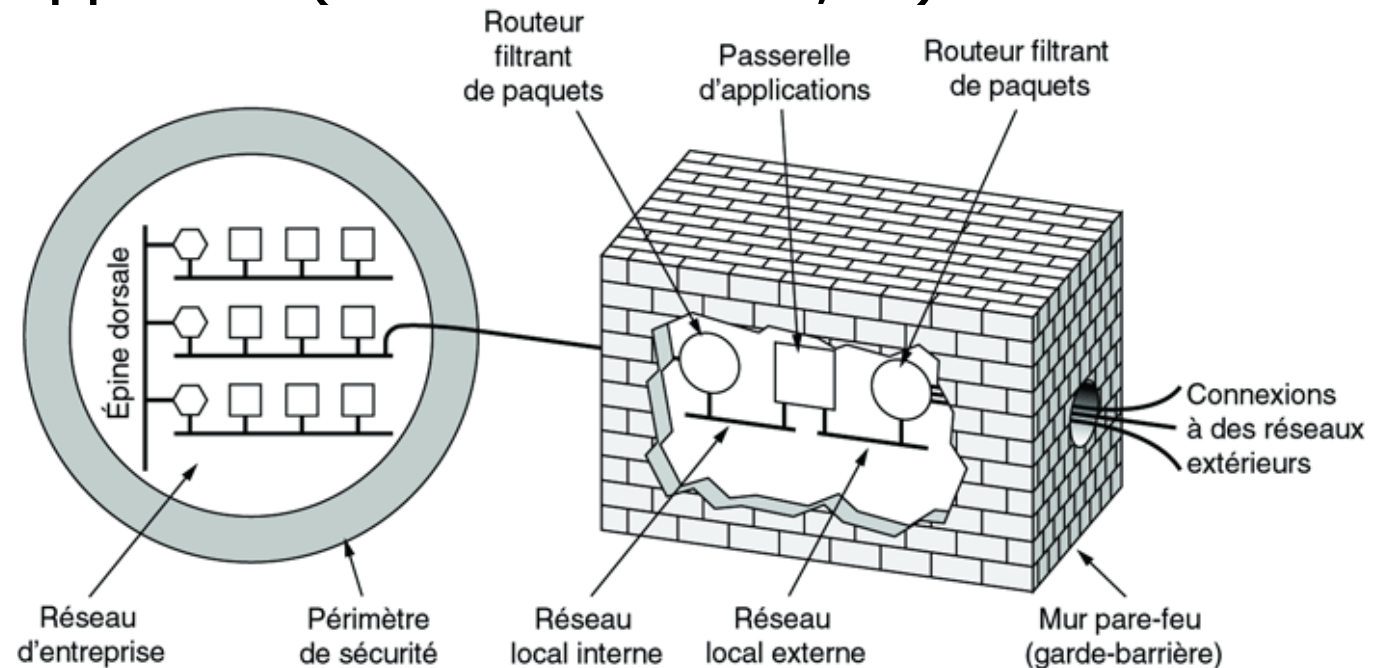
- Réseaux source et destination de même type (paquets IP par ex.) séparés par un réseau longue distance de type ATM par ex.



© Pearson Education France

# Parefeu (garde-barrière ou *firewall*)

- Routeurs filtrants
  - filtrage des paquets par @ source/destination et/ou port
  - routeur interne : filtrage des paquets sortants
  - routeur externe : filtrage des paquets entrants
- Passerelle applicative
  - filtrage applicatif (contenu de mail, ...)



# Réseaux privés virtuels (VPN)

- Soucis de sécurité pour une organisation
- Deux solutions
  - a) liaisons louées entre les sites mais coûteux
  - b) VPN : interconnexion des sites par un tunnel négocié entre deux garde-barrières avec cryptage et authentification

