

Les couches transport UDP et TCP

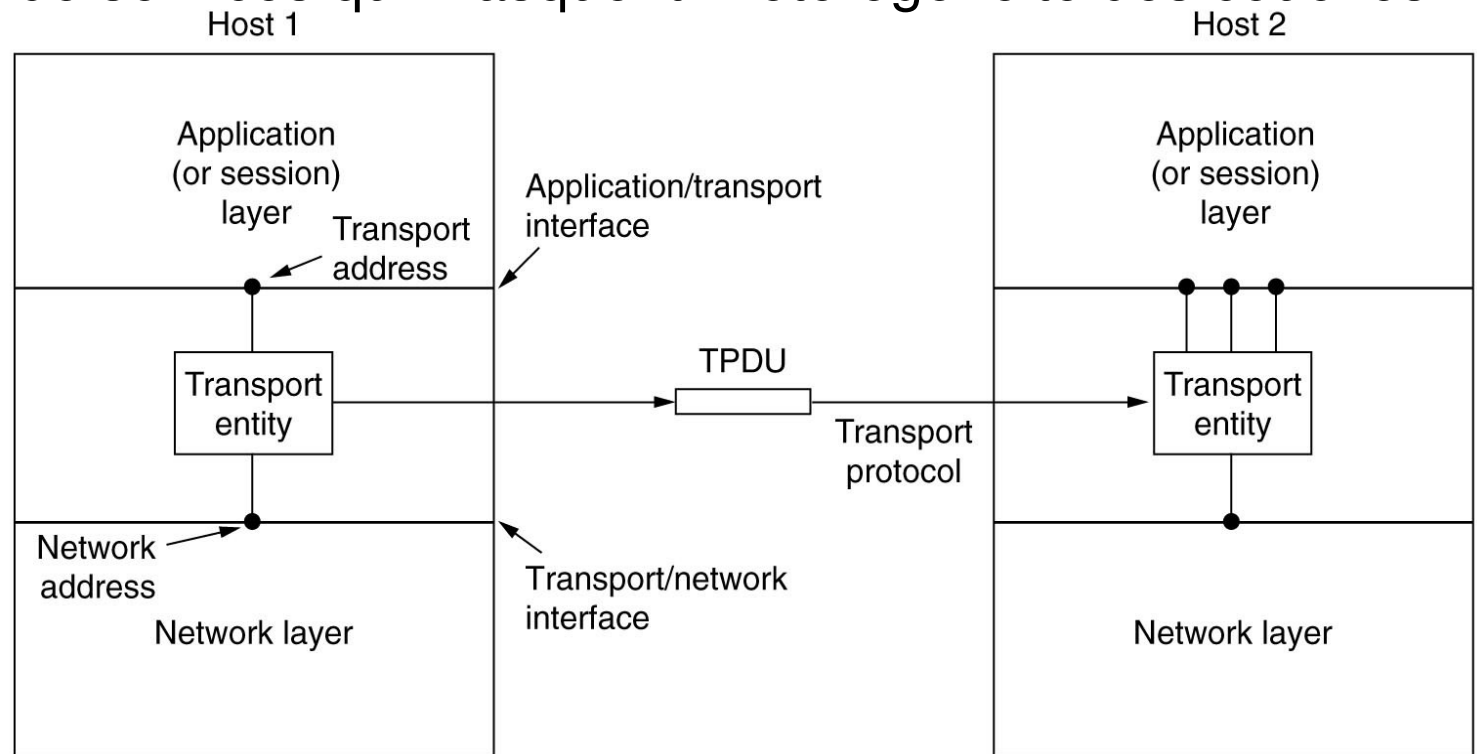
- Objectifs et plan:
 - Couche 4 : plaque tournante entre
 - Fournisseurs des services de transport (les « réseaux »)
 - Utilisateurs des services de transport (les « applications »)
 - Des primitives de transport
 - Les sockets
 - Deux « services »
 - Sans connexion, non fiable : UDP
 - Orienté connexion, fiable : TCP

Bibliographie et sources

- **Les cours de Stéphane Lohier** <http://igm.univ-mlv.fr/~lohier>
- **CCNA ICND1 et CCENT**, 2° ed., Wendell Odom Pearson Education 2007.
- **Réseaux**, 4^{ème} éd. Andrew Tanenbaum, Pearson Education 2003.
- **Analyse structurée des réseaux**. Kurose & Ross, Pearson Education 2003.
- **Java et Internet** Roussel, Duris, Bedon et Forax. Vuibert 2002.
- **Les réseaux**, 3^{ème} éd. Guy Pujolle, Eyrolles 2000.
- **Interconnections**, 2nd ed. Radia Perlman. Addison Wesley 2000.

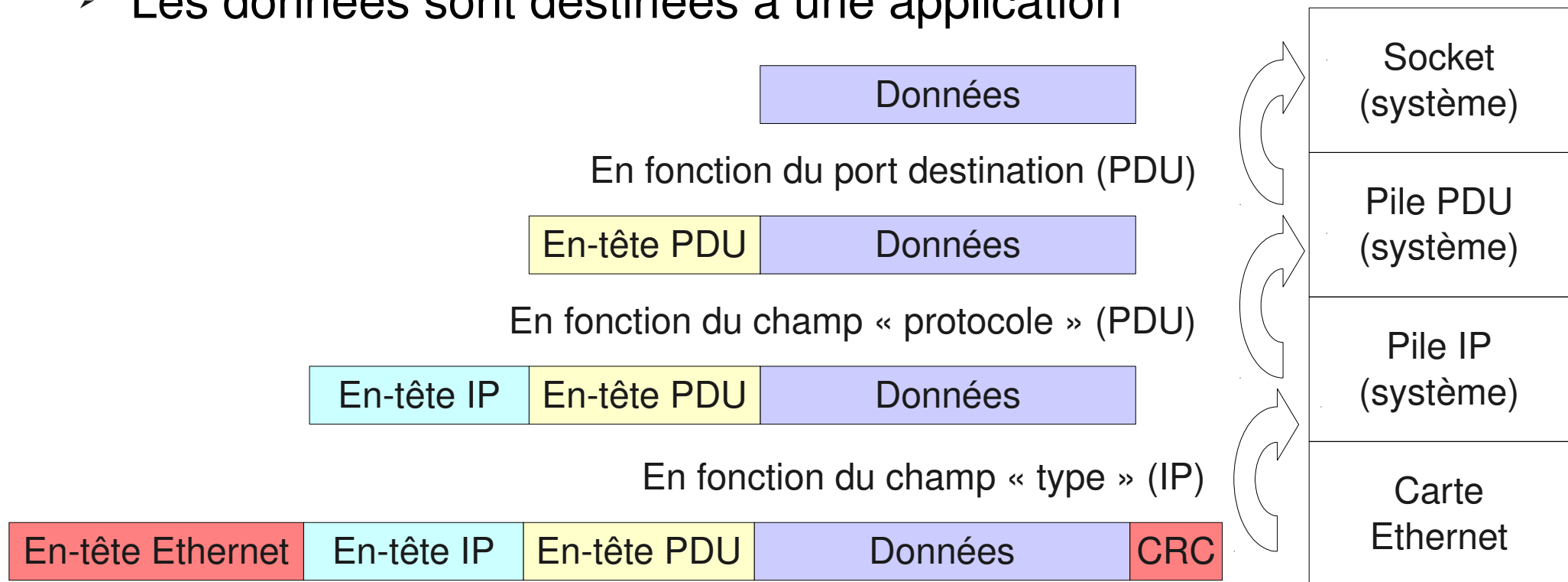
La couche transport (4)

- **Fournisseurs du service transport (1-2-3-4)**
 - Assurer la transmission et l'acheminement des données
- **Utilisateurs du service transport (« applications »)**
 - Disposer de services qui masquent l'hétérogénéité des couches basses



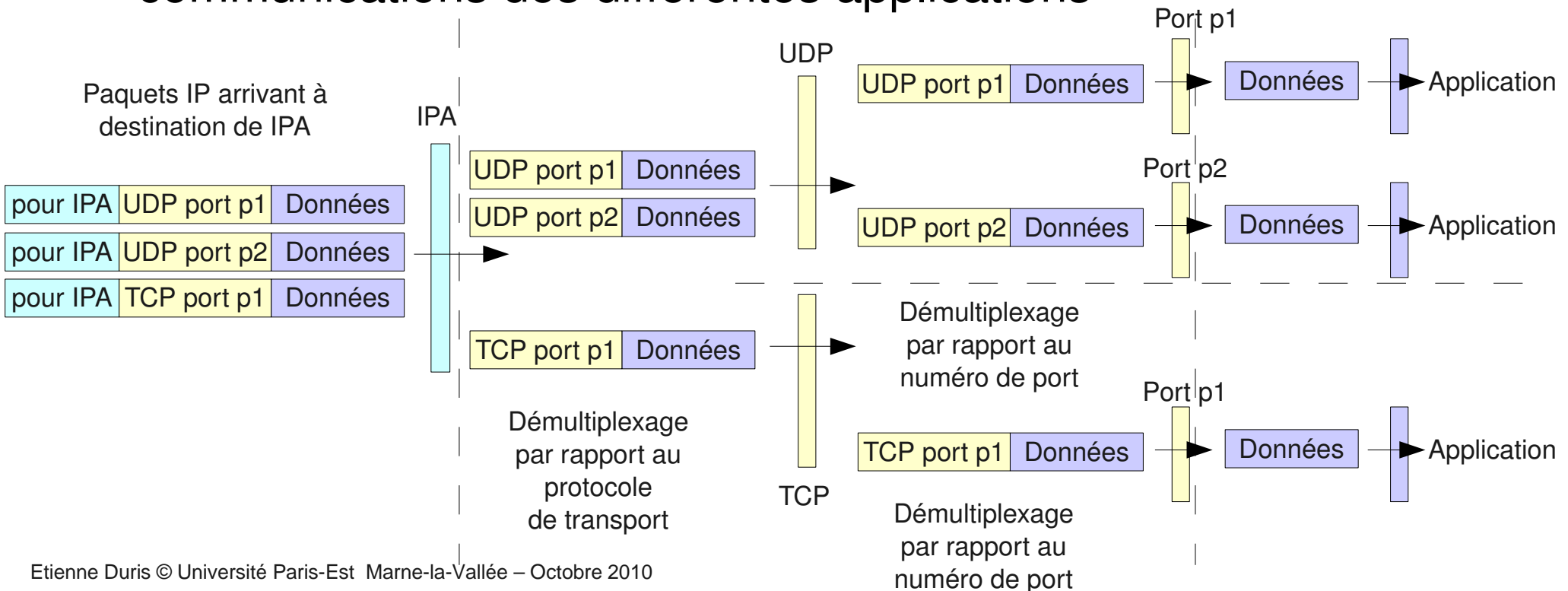
Encapsulation des données

- Une trame Ethernet est adressée à une adresse MAC
- Un paquet IP (couche 3) est destiné à une adresse IP
- Un PDU (couche 4 – UDP ou TCP) est destiné à un port
- Les données sont destinées à une application



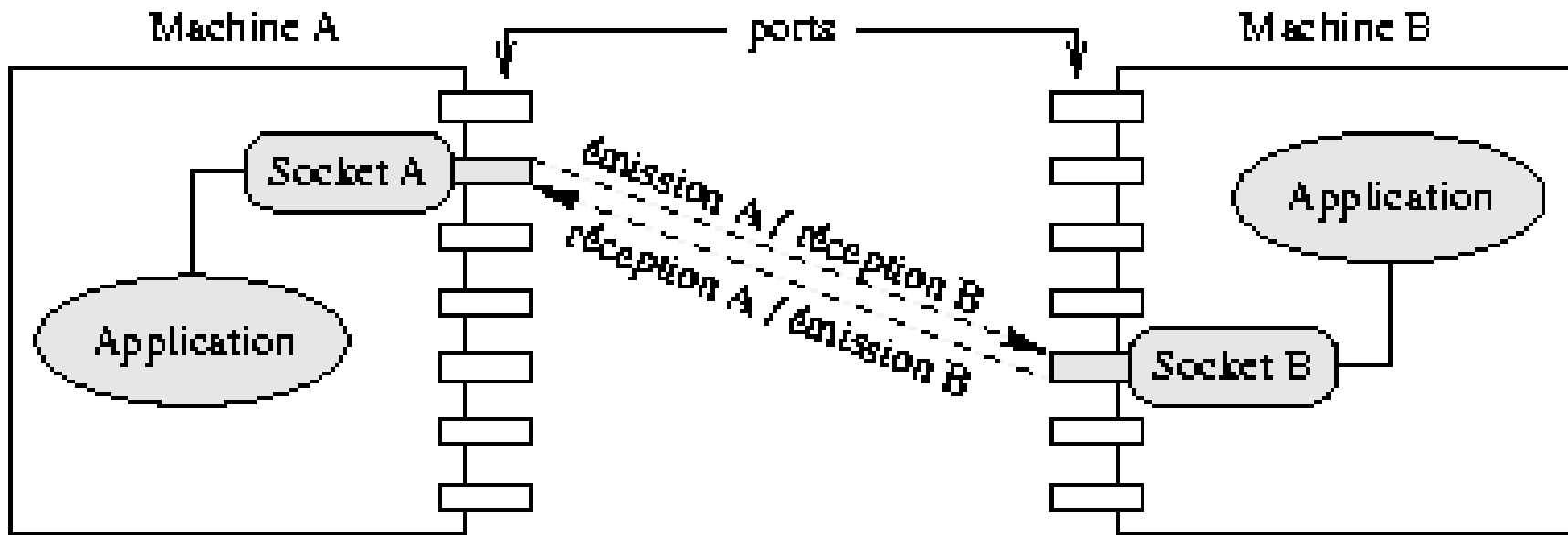
Multiplexage / démultiplexage

- Les applications se lient à des **sockets** pour accéder au réseau
- Ces sockets système sont identifiées par une **adresse IP** et un **numéro de port**
- Elles permettent de multiplexer/démultiplexer les communications des différentes applications



Le rôle des sockets

- Pour UDP et TCP, les sockets jouent le même rôle et sont identifiées par une adresse IP et un numéro de port.
- Au moins 2 sockets sont impliquées dans une communication

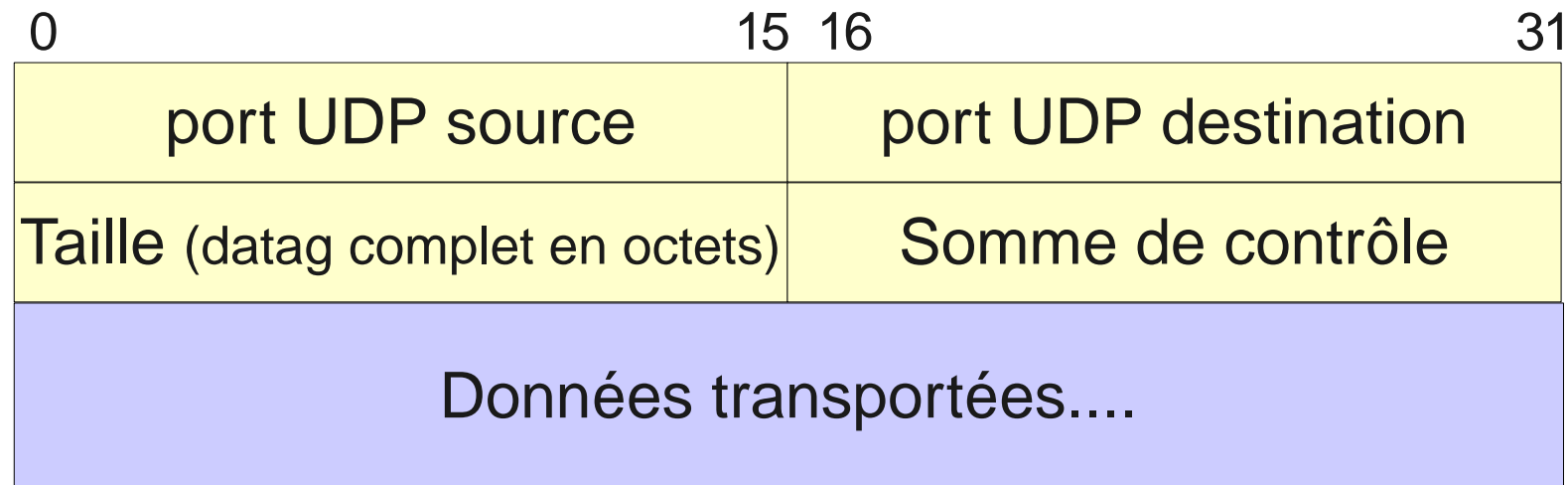


Le protocole UDP

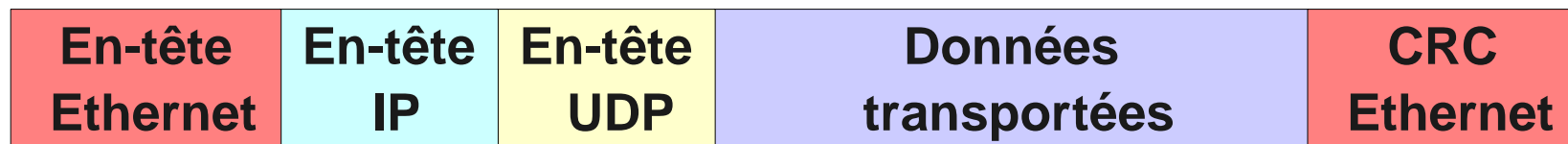
- **User Datagram Protocol** (RFC 768)
 - acheminement de **datagrammes** ou **segments** au dessus de IP
 - **pas de fiabilité** supplémentaire assurée
 - assure la **préservation des limites** de chaque datagramme
- Le multiplexage/démultiplexage au niveau des machines se fait *via* la notion de port (comme en TCP)
 - certains ports sont affectés à des services particuliers
 - RFC 1700 ou www.iana.org/assignments/port-numbers
 - En général, les n° de port inférieurs à 1024 sont réservés (*root*)

Format des segments UDP

- Taille maximale des données transportées: $(2^{16}-1-8) \sim 64\text{Ko}$
- Checksum optionnel en IP v4, obligatoire en IP v6
 - ajout (pour le calcul) d'un pseudo-en-tête avec @ip dest et @ip src



- Exemple d'encapsulation dans une trame Ethernet:



Caractéristiques et utilisations d'UDP

- UDP est léger mais n'offre que peu de services
- Échanges simples de messages courts client/serveur
 - DNS (*Domain Name Service*)
 - Question: « [Quelle est l'adresse de google.fr?](#) »
 - Réponse: « [google.fr est à 72.14.221.104](#) »
 - RPC (*Remote Procedure Call*)
 - Appel de procédure avec paramètres / retour du résultat
- Simple, donc léger, donc à priori relativement rapide
 - RTP (*Real Time Protocol*) pour multimédia, audio, vidéo...

Communication en diffusion

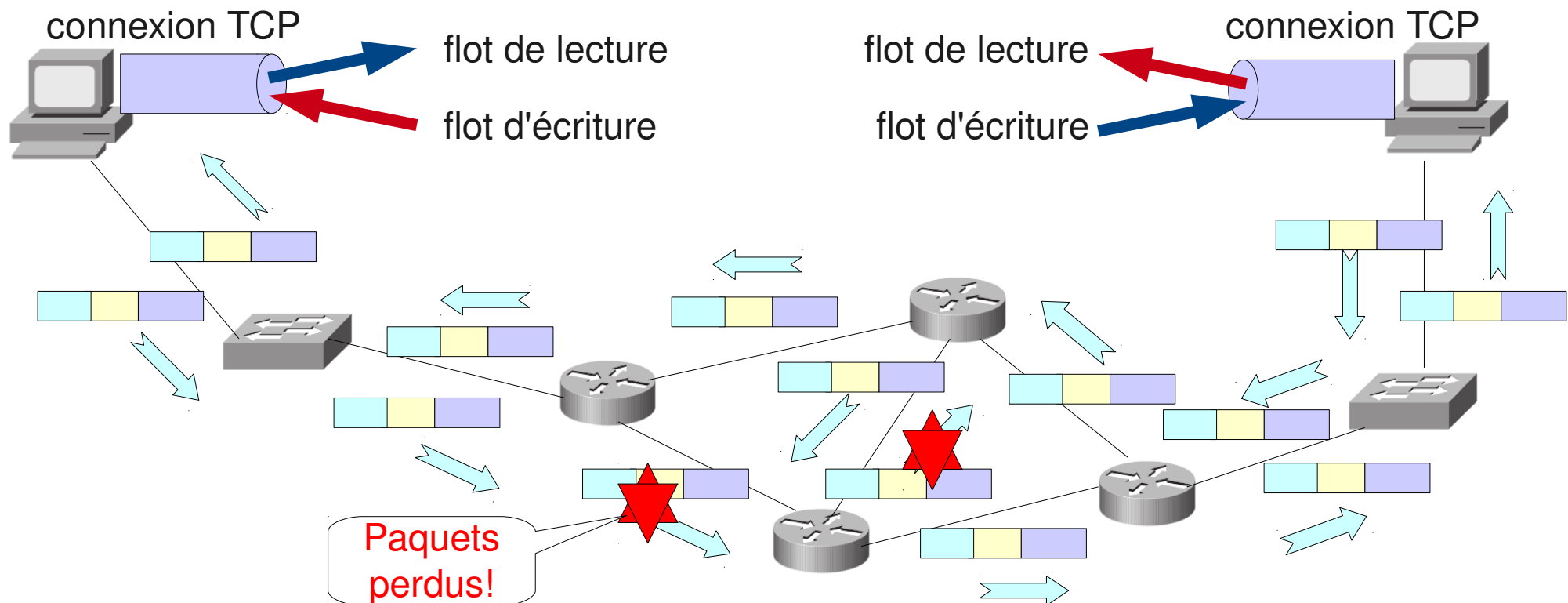
- UDP non connecté: autorise les **communications en diffusion**
 - Efficace lorsque le protocole de liaison de données sous-jacent offre la diffusion (ex. Ethernet):
 - **Une seule trame** peut être utilisée pour transporter un segment destiné à **plusieurs machines**
 - **Broadcast** : de un à tous, un envoi à l'adresse IP 255.255.255.255 est destiné à toutes les machines du réseau local
 - **Multicast** : de un à plusieurs, un envoi à une adresse IP de classe D (224.0.0.0/4) est destiné à toutes les machines qui se sont explicitement « **abonnées** » à ce **groupe de diffusion**
 - Nécessite un mécanisme pour l'abonnement et le routage (IGMP)
 - L'adresse IP de classe D doit être associée à un port UDP pour constituer un groupe de diffusion de données

Le protocole TCP

- *Transmission Control Protocol*, RFC 793
 - Communication
 - **Par flots**: ne préserve pas les limites mais l'ordre
 - **Fiable**: les données perdues par le réseau sont réémises
 - **Mode connecté**: on ne peut communiquer qu'avec une machine OK
 - **Full duplex**: deux canaux bidirectionnels sont disponibles
 - Données bufferisées, encapsulées dans des paquets IP
 - flot découpé en segments (~536 octets)
 - Mécanismes de contrôle de flot
 - ex: contrôle de congestion
 - assez lourd d'implantation (beaucoup plus qu'UDP)

Connexion virtuelle

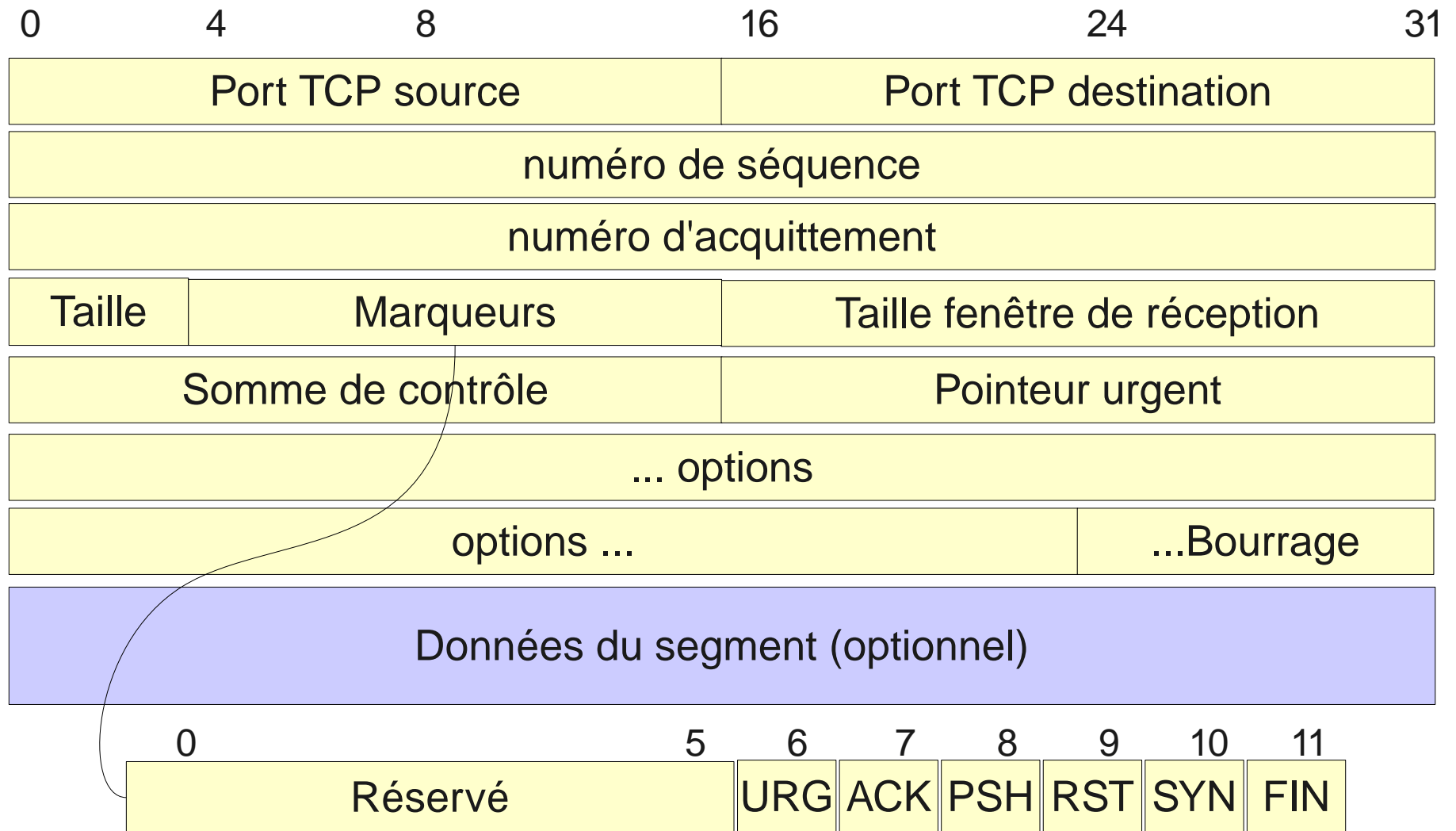
- L'implémentation de TCP donne l'illusion d'une **connexion bidirectionnelle** établie entre les 2 machines, même si ce n'est que **virtuel**
 - L'acheminement est réalisé par routage de paquets IP



Principe des segments

- La fiabilité est obtenue par un mécanisme d'acquittement des segments
 - À l'émission d'un segment, une alarme est amorcée
 - Elle est désamorcée quand l'acquittement correspondant est reçu
 - Si elle expire, le segment est réémis
- Chaque segment possède un numéro de séquence
 - Pour préserver l'ordre, éviter les doublons
 - Les acquittements sont identifiés par un marqueur ACK
 - Transport dans un même segment des données et de l'acquittement des données précédentes: piggybacking

Format de trame TCP (en-tête)



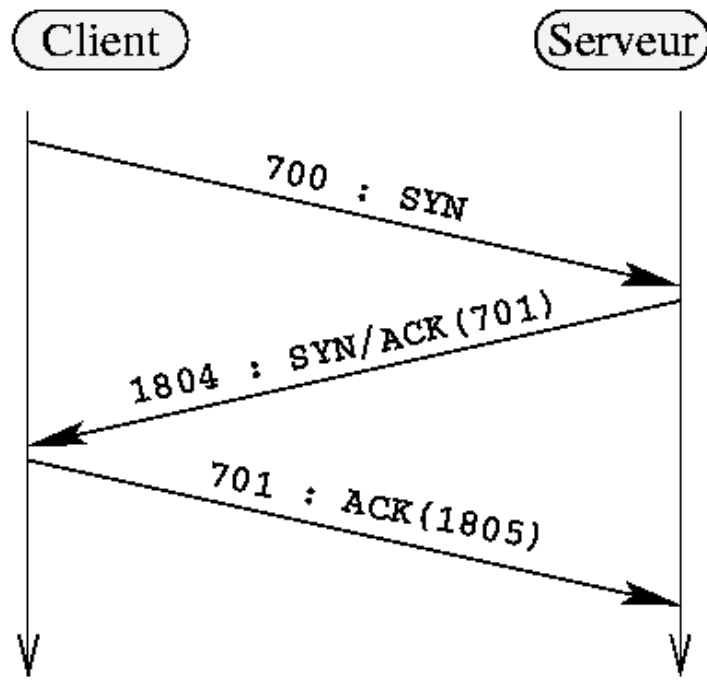
Format (suite)

- Champ Taille
 - en nombre de mots de 32 bits, la **taille de l'en-tête**
- Marqueurs
 - URG données urgentes (utilisation conjointe pointeur urgent)
 - ACK acquittement
 - PSH force l'émission immédiate (w.r.t. temporisation par défaut)
 - RST refus de connexion
 - SYN synchronisation pour la connexion
 - FIN terminaison de la connexion
- Somme de contrôle (comme IP v4 avec un pseudo en-tête)
- Options
 - Exemple: taille max. de segment, estampillage temporel

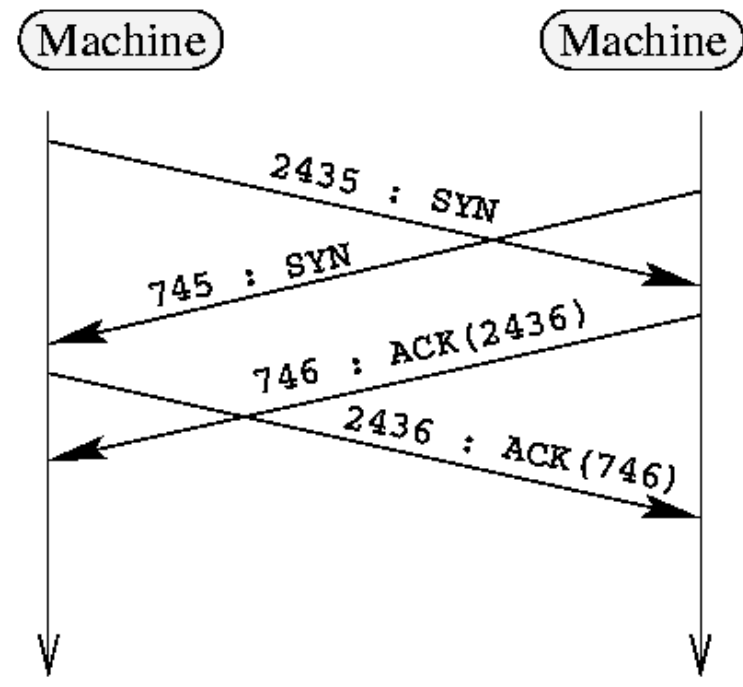
Établissement de la connexion

➤ *Three Way Handshake*

- SYN ; SYN/ACK ; ACK



Connexion client-serveur

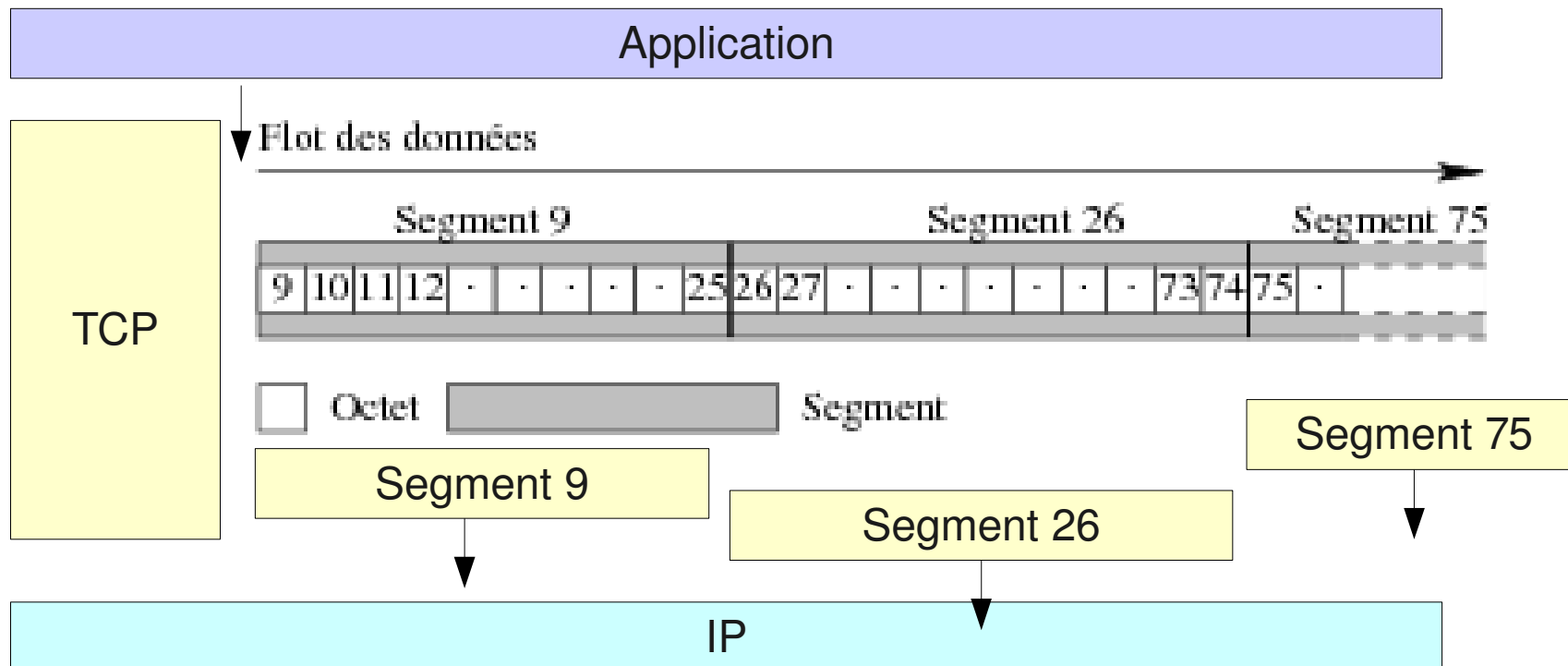


Connexion simultanée

- Refus de connexion: SYN --- RST/ACK

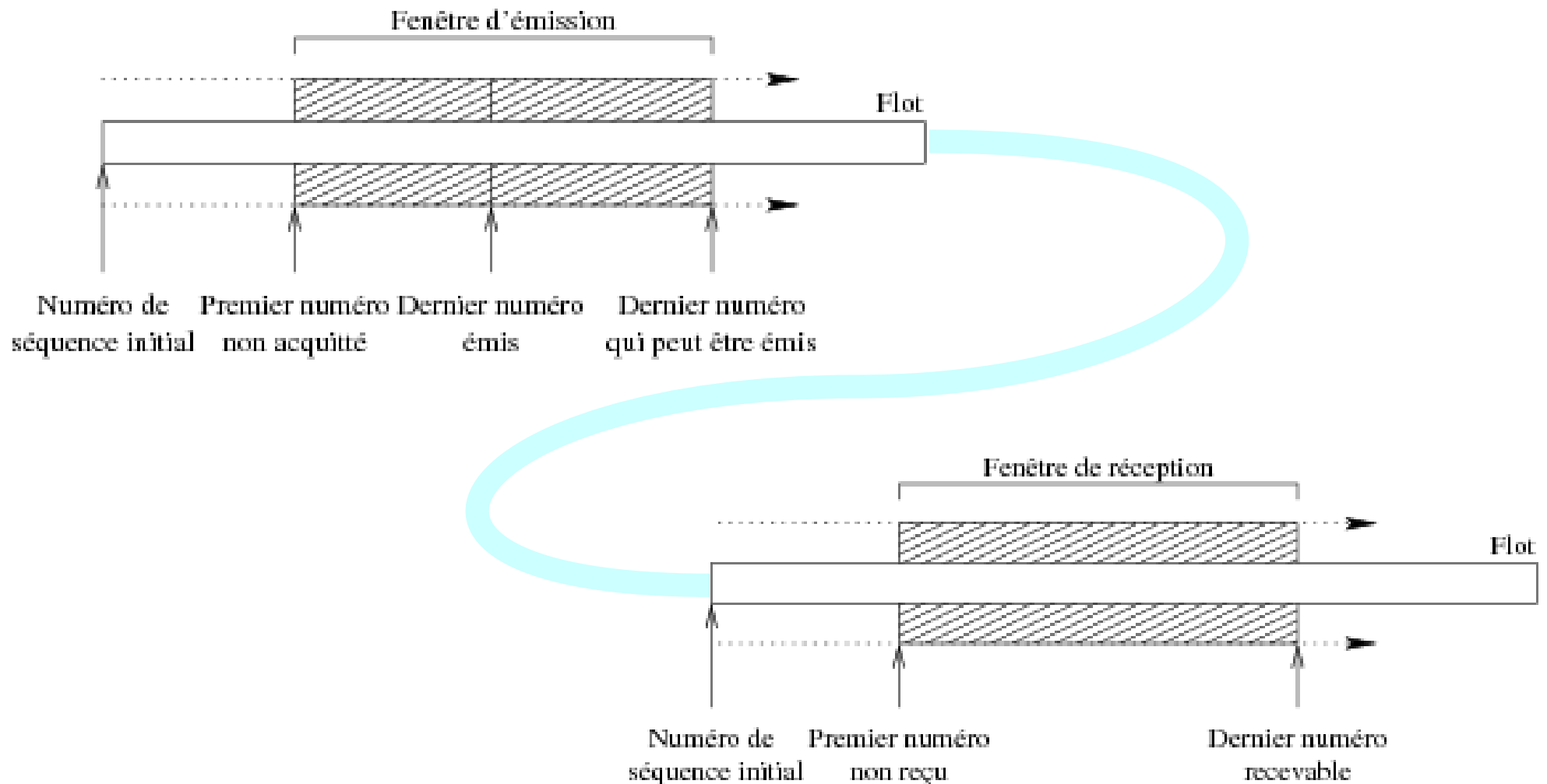
Gestion des transmissions

- TCP **segmente** le flot de données qui lui est transmis par l'application de sorte à confier « morceau par morceau » ces segments à la couche réseau (IP)
- Chaque segment est numéroté par le numéro de séquence de son premier octet dans le flot



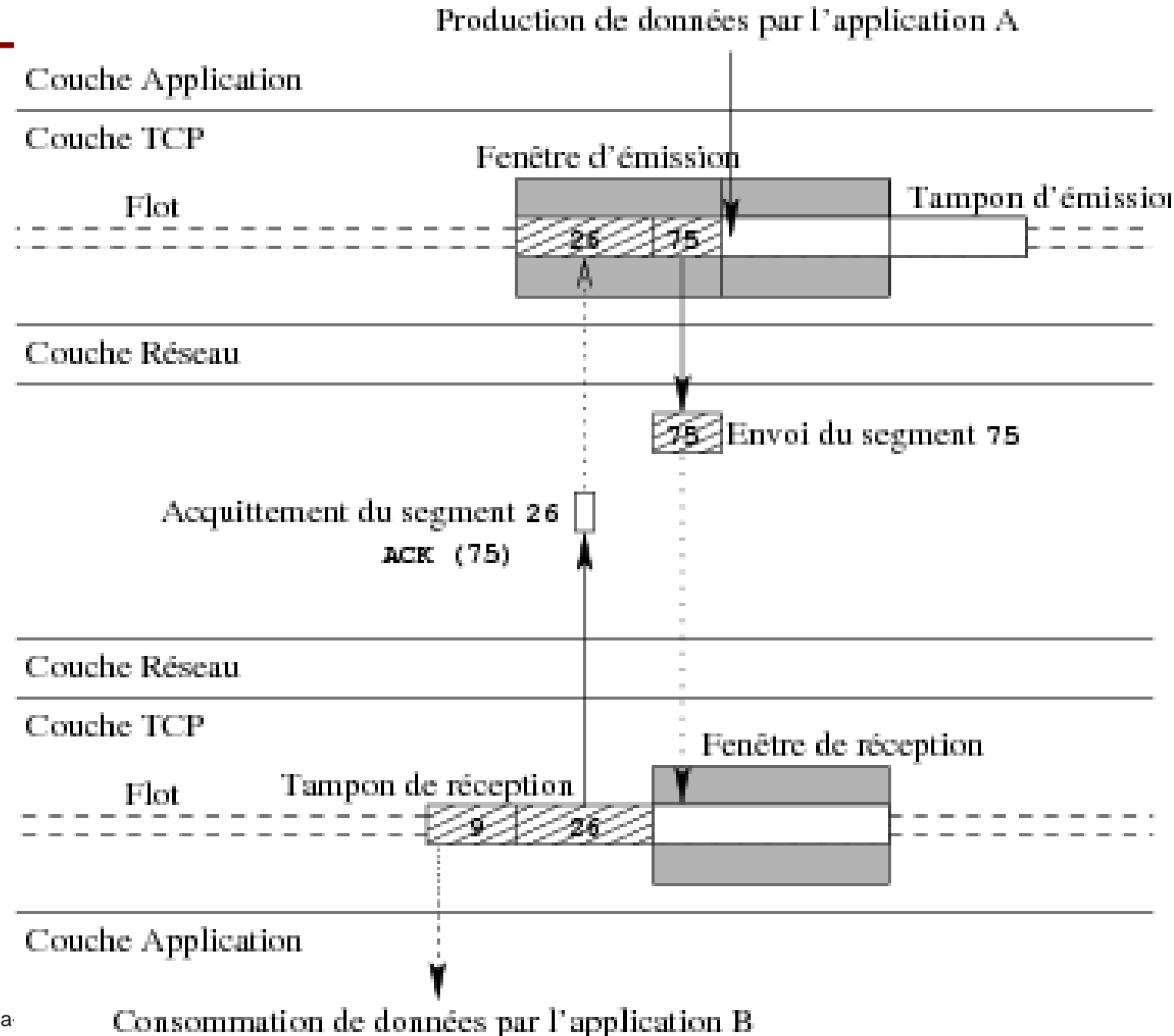
Fenêtres d'émission et de réception

- Dans chaque sens de la communication full duplex, il y a 2 fenêtres qui délimitent ce qui est respectivement reçu et envoyé



Acquittement et contrôle de flot

- Exemple simplifié d'utilisation des tampons et des fenêtres d'un flot d'une machine A vers une machine B.



Libération d'une connexion

- La déconnexion nécessite l'accord des 2 extrémités
 - Possibilités de *half-closed*
 - Temps d'attente (2 fois la durée de vide d'un segment)
 - Interdit la réutilisation rapide des ports TCP

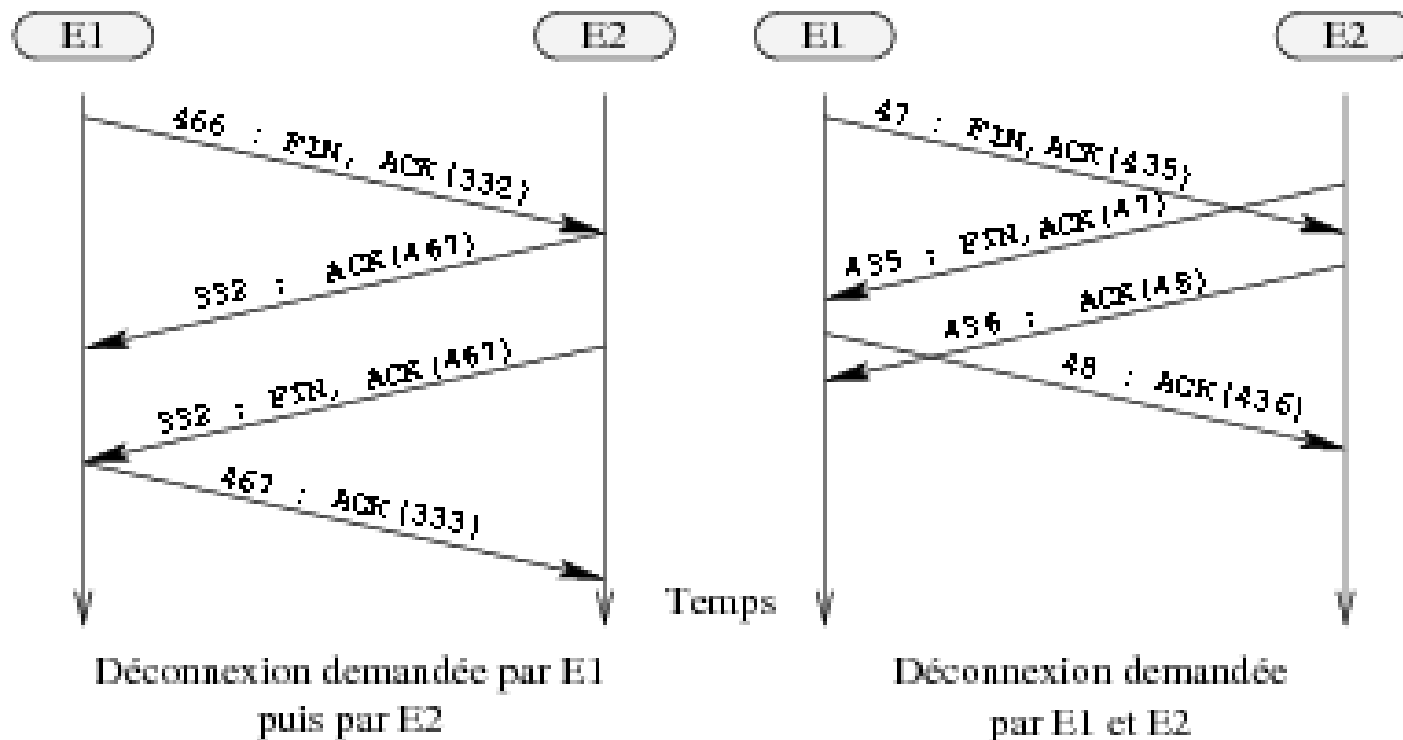
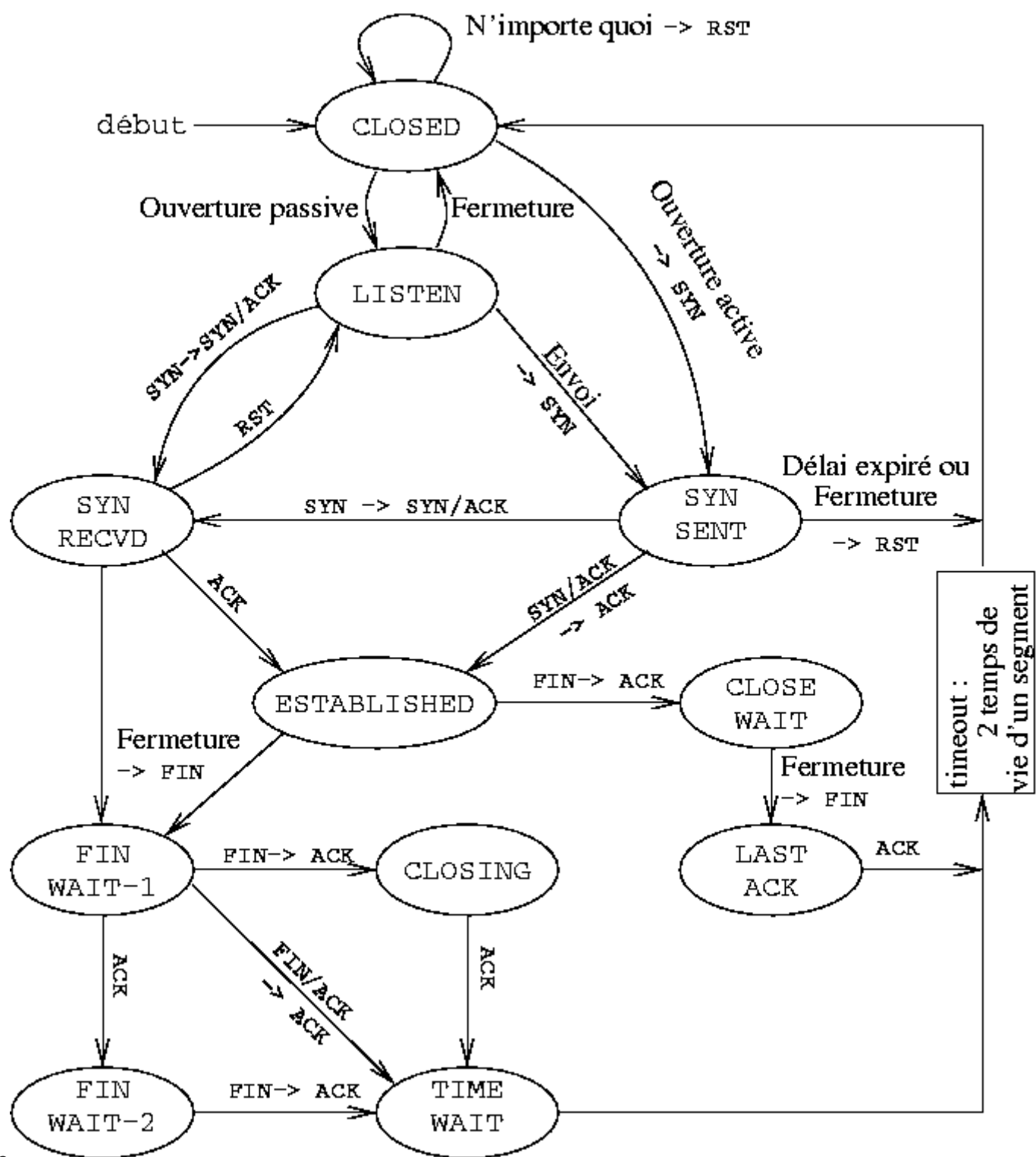


Diagramme d'états



L'importance des délais dans TCP

- Le concept même d'acquittement impose des notions de délai
- Par exemple, quel est le délai au delà duquel un segment non acquitté doit être réémis
 - Le RTO (*Retransmission TimeOut*) est calculé à partir d'un temps d'aller et retour moyen RTT (*Round Trip Time*)
- On veut aussi pouvoir assurer qu'un segment ne « séjourne » pas trop longtemps dans le réseau
 - Par exemple pour éviter les doublons (segment réémis car pas acquitté à temps)
 - Champ TTL des paquets IP
 - Estampillage temporel des paquets