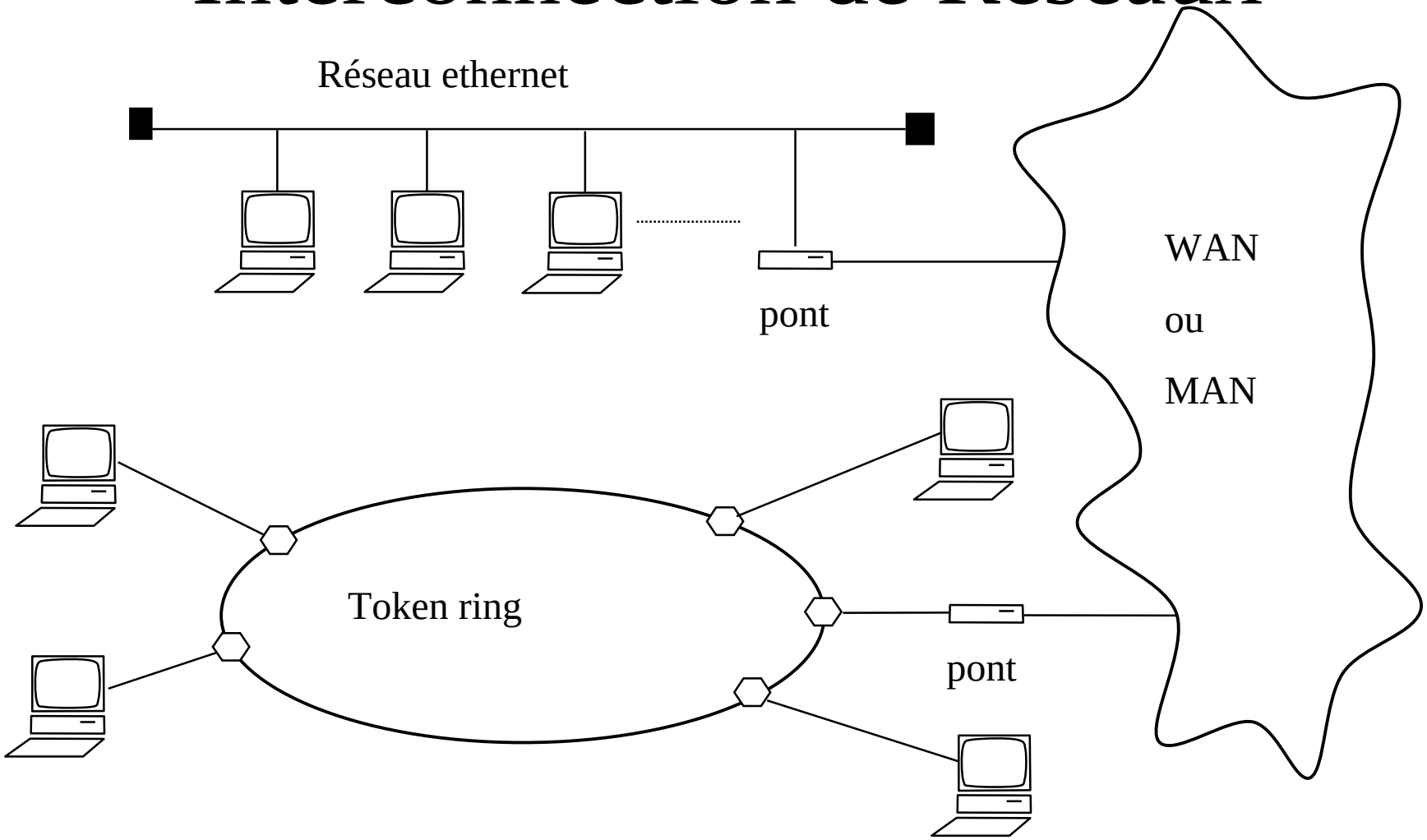
Systèmes informatiques
Protocole INTERNET
Communication par socket avec UDP et TCP

G.Berthelot

ENSIIE 1A 2009

Interconnection de Réseaux



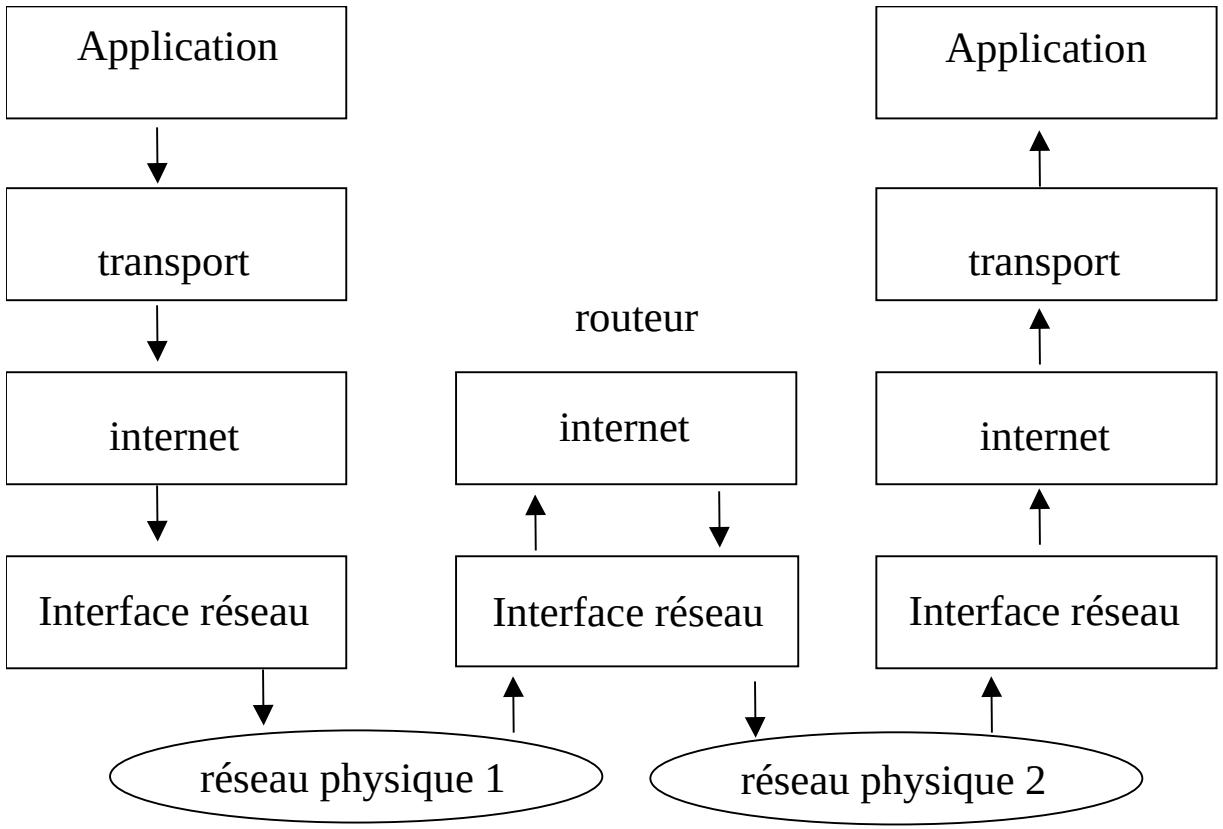
Problèmes de transfert inter-réseaux

- référentiels d'adresses différents
- longueurs de trames différentes
- taux d'erreurs différents
- comment faire le routage sur les noeuds intermédiaires ?
- comment faire le contrôle de flux ?

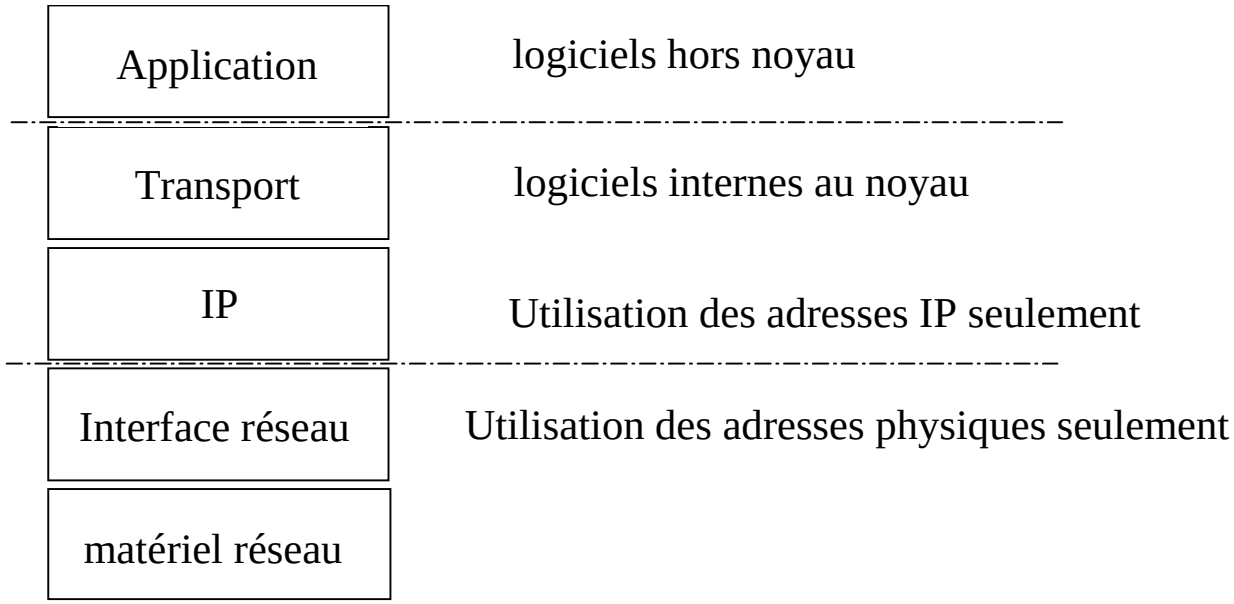
Internet Protocol

- *Internet Protocol permet de rassembler un ensemble de réseaux physiques interconnectés par des passerelles au sein d'un réseau "virtuel".*
- *Chaque station possède une adresse IP*

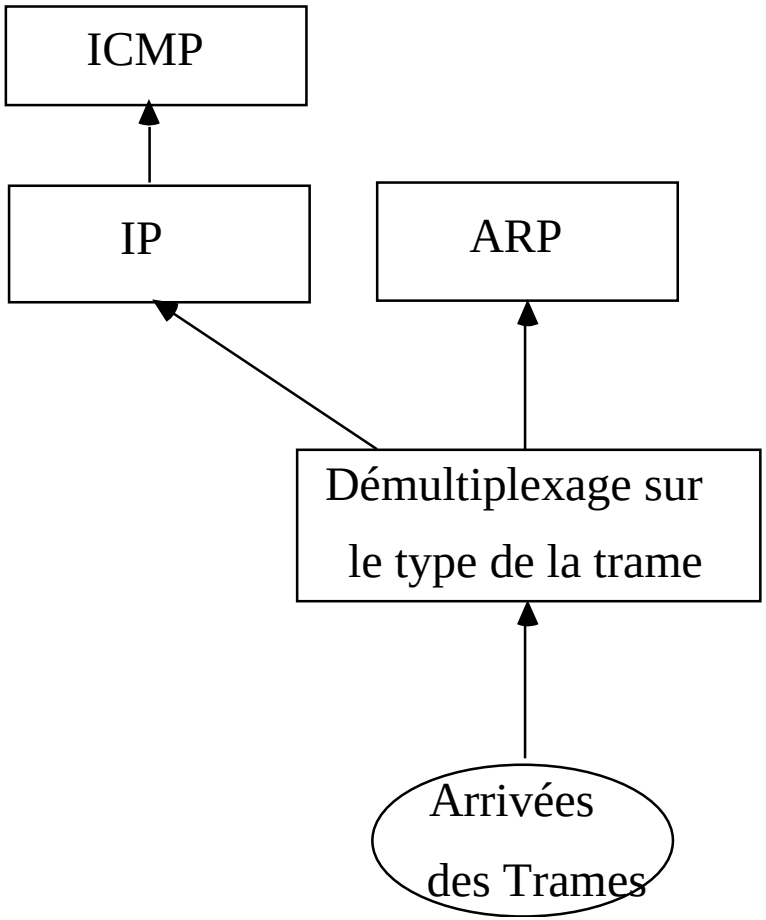
Situation du protocole internet dans les piles de protocoles



Insertion de la pile internet dans le système d'exploitation



répartitions des paquets



Protocole IP

fonctionalités

- Adressage Internet et Routage entre Réseaux
- Fragmentation/Réassemblage, Adaptation de la taille des messages soumis par la couche Transport suivant les possibilités offertes par la couche Liaison (MTU).
- Communications dans le mode minimal : **DATAGRAMME** (mode non connecté), Envois de paquets **sans acquittements**

⇨ la détection des messages erronés ou perdus et leurs réémissions sont à la charge de l'émetteur des messages (couche Transport).

- Multiplexage/Démultiplexage par rapport à la couche Transport

adressage IPv4 3

adressage uniforme quels que soient les sous réseaux traversés

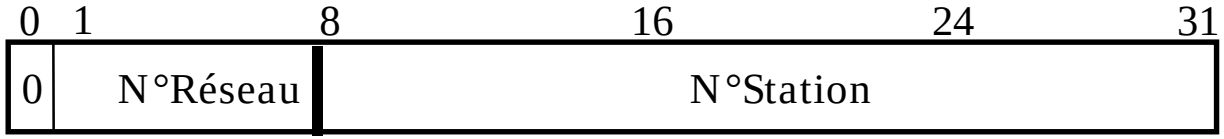
⇨ les adresses sont uniques et universelles.

⇨ couple (N° de réseau, N° de station sur le réseau)

Adresse sur 32 bits soit 4 octets : notation pointée T.U.V.W

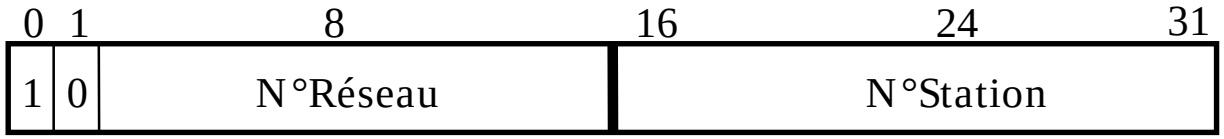
Classes d'adresses

Classe A : Peu de réseaux de cette classe, de nombreuses stations par Réseau



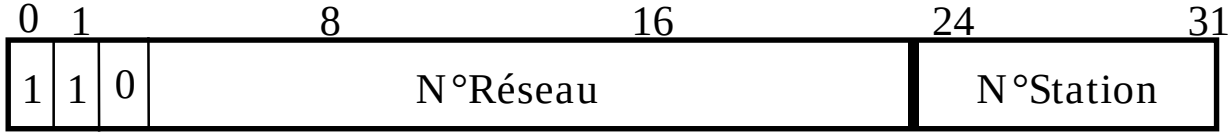
N°de Réseau : 1 - 126
127 désigne l'adresse locale pour le **rebouclage**

Classe B :



N°de Réseau : 128.1 - 191.254

Classe C : Beaucoup de Réseaux, Peu de Stations par Réseau
La classe la plus répandue

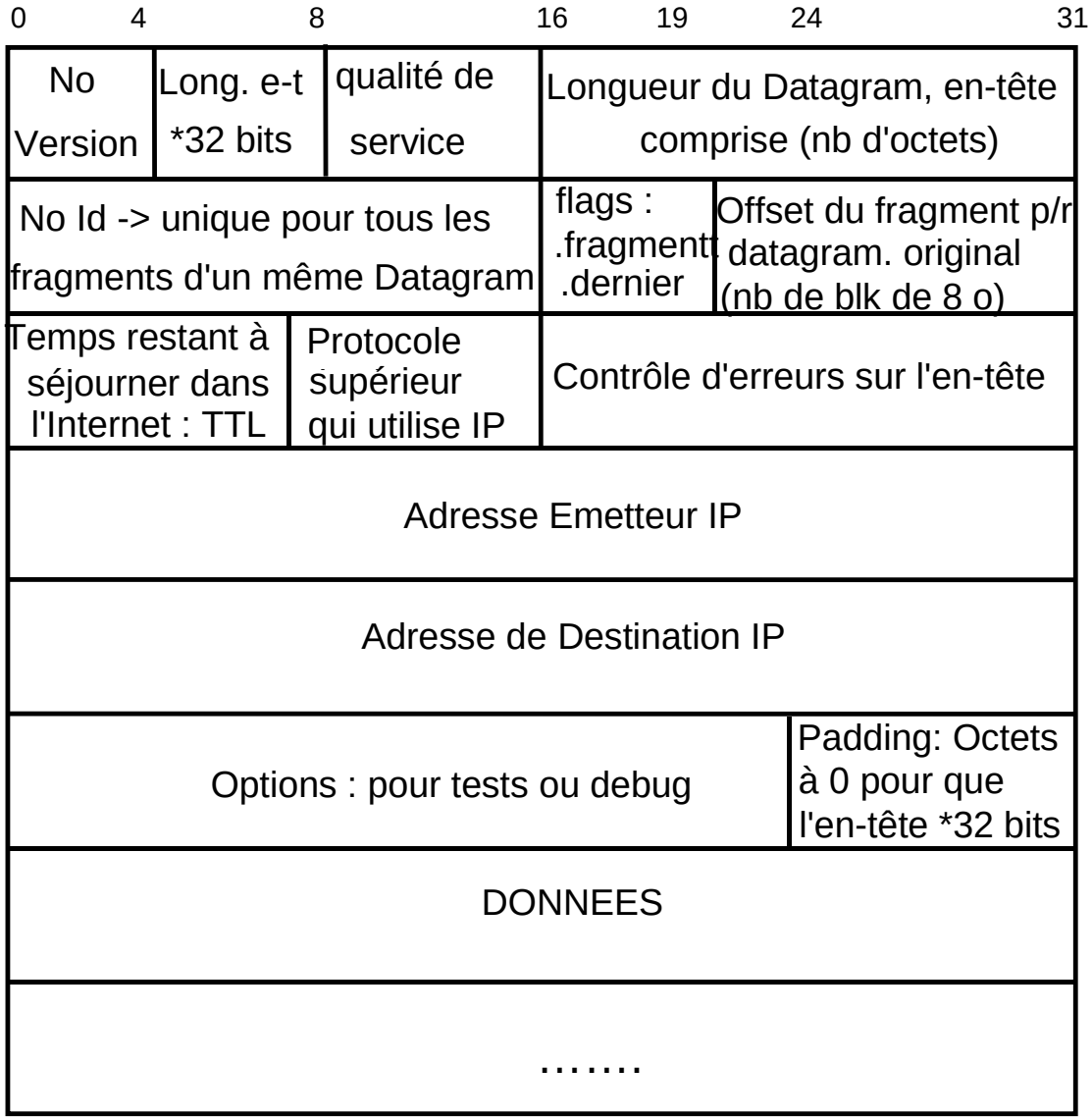


N°de Réseau : 192.0.1 - 223.255.254 N°de Station : 1 - 254

Remarques

- Adresse réseau : l'adresse du réseau est obtenue en mettant un 0 dans le champs n° station
- Adresse courante : une adresse ne comportant que des zéros indique la station courante
- Adresse de rebouclage : pour permettre à deux applications sur le même site de communiquer : 127.0.0.1
- Adresse de diffusion (broadcast) : Pour diffuser un paquet à toutes les stations d'un réseau on utilise une adresse spéciale obtenue en mettant à 1 tous les bits du champs n° station (soit 255 pour un réseau de classe C)
- possibilité de sous-adressage (plusieurs "sous réseaux dans un réseau) avec un netmask (un ET bit à bit entre le netmask et l'adresse IP donne le numéro du réseau)

Format d'un datagramme IP



Principaux problèmes IPv4

- Tarissement des adresses.
- Grossissement des tables de routage.
- Trop grande centralisation des distributions d'adresses.

Le plan d'adressage internet IPv4 doit tôt ou tard arriver à saturation

Ces difficultés (et d'autres) ont amené à spécifier une version nouvelle.

IPv6: critères de conception

- Grand espace d'adressage hiérarchisable.
- Autorisant un routage hiérarchisé.

  => Diminution des tailles des tables
- Distribution d'adresses facilitée en répartissant les possibilités d'attribution.
- Support de mécanismes de sécurité (confidentialité, authentification ...)
- Support de la qualité de service (données multimédia, sans pertes, ...).
- Support du mode diffusion.
- Support d'artères à tous les débits (de très faible à très élevé).

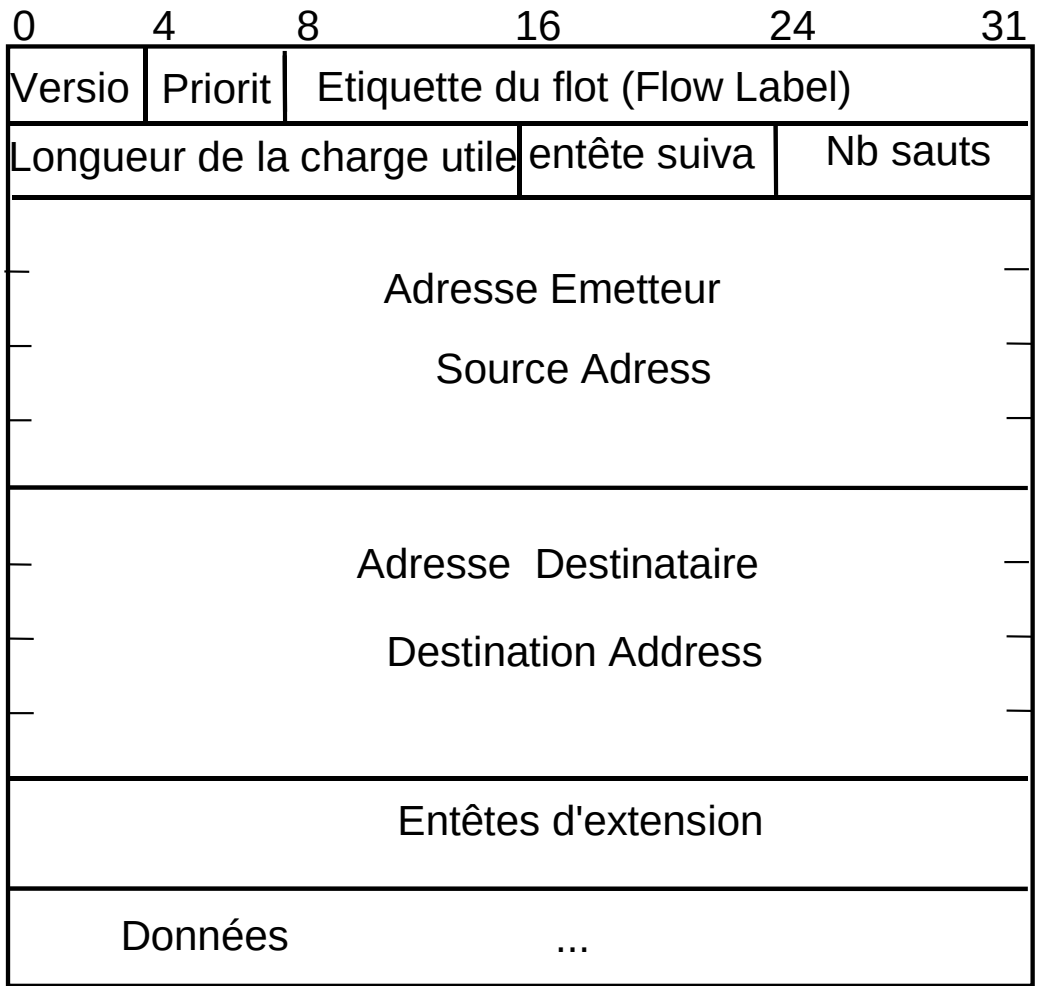
IPv6: critères de conception

- Capacité d'adressage quadruplée

  128 bits soit 16 octets (~10^{38} adresses).
- Simplification du format d'en-tête standard

  (en-tête minimum plus extensions eventuelles)
- Optimisation pour un routage simplifié.
- Suppression des champs inutiles au routage.
- Alignement sur des frontières de mots
- Identification de flots d'octets par Etiquette pour permettre la réservation de ressources routeur => qualité de service
- Pas de somme de contrôle d'en-tête

IPv6: format du paquet



Protocole ICMP
(Internet Control Message Protocol)

protocole de supervision : mécanisme de communication entre la couche IP d'une station et la couche IP d'une autre station :

- demande d'écho (ping)
- réponse à une demande d'écho
- destination inaccessible
- limitation de production de la source
- expiration du TTL d'un datagramme
- .....

Protocole ARP
(Adress Resolution Protocol)

Problème
Une station doit envoyer un paquet IP à une station appartenant au même réseau auquel elle appartient. Elle doit encapsuler ce paquet dans une trame qui sera envoyée à la machine destinataire.

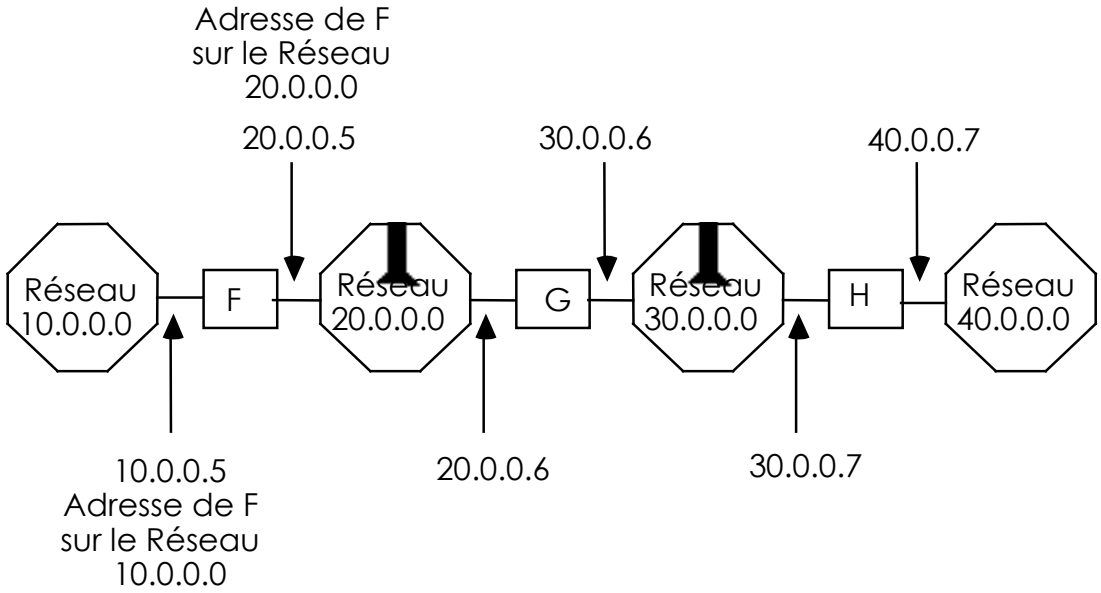
A quelle adresse physique cette trame doit-elle être envoyée ?

ARP permet de trouver l'adresse physique d'une station à partir de son adresse IP en diffusant un message "A qui est attribuée cette adresse IP ?" (connaissance du netmask nécessaire pour diffusion)

Si pas de réponse la station est inacessible !

protocoles inverses : RARP, BOOTP, DHCP

IPv4 : routage



Pour envoyer à un site sur le Réseau suivant	Envoyer à cette Adresse
20.0.0.0	Délivrer Direct
30.0.0.0	Délivrer Direct
10.0.0.0	20.0.0.5
40.0.0.0	30.0.0.7

Table de Routage de G

Politiques de routage

Les schémas décrits plus haut correspondent à un routage statique créé par configuration ou par commande d'administration.

Le **routage dynamique** ne change rien à la façon dont les tables de routages sont utilisées dans la couche IP. Les informations qui sont contenues dans les tables sont modifiées au fur et à mesure de l'évolution du réseau.

RIP (Routing Information Protocol) :

Dans un routeur, le routage est calculé par rapport à la plus courte distance vers un destinataire, la distance est comptée en nombre de noeuds traversés (hop).

Exemple : toutes les 30 s :

- Un routeur transmet une demande de table de routage vers tous ses voisins.
- Chaque voisin transmet sa table de routage.
- Après réception de toutes les tables, le routeur recalcule sa table de routage.

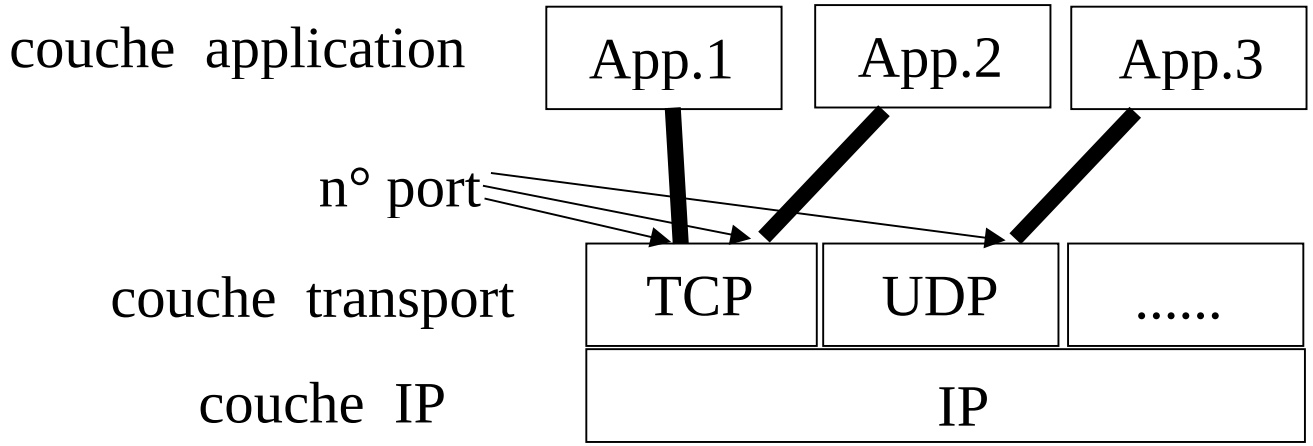
Politiques de routage

OSPF (Open Shortest Path First): Dans un routeur, le routage est calculé en fonction des informations d'état des liens entre noeuds. Un **coût est associé à un lien**. Il peut être : le débit, le délai de propagation, ...

Les informations doivent être échangées entre routeurs pour maintenir une connaissance "cohérente" de l'état du réseau.

La couche transport : TCP - UDP

- Les point d'accès aux services offerts par cette couche sont désignés par des "extrémités".
- Une extrémité est l'association d'une adresse IP et d'un numéro de port
- un n° port distingue une application parmi l'ensemble des applications accessibles à travers la couche transport d'une machine distante



Protocole User Datagram Protocol (UDP)

fonctionalités :

le protocole UDP fournit un service non fiable et sans connection utilisant les adresses IP pour transporter des messages entre stations. Il ajoute la capacité de distinguer parmi de multiples destinations à l'intérieur de la station destinataire.

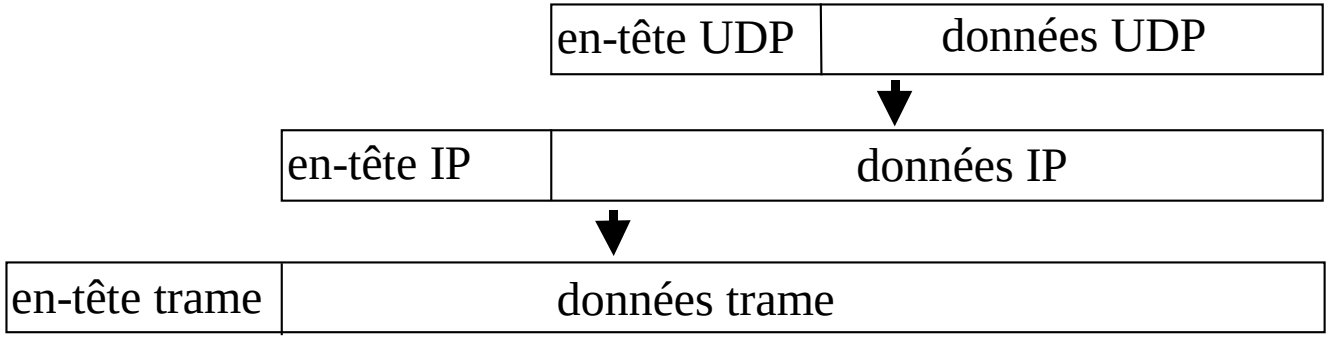
les messages UDP (user datagram) peuvent
- être perdus
- être dupliqués
- arriver dans le désordre

pas de contrôle de flot

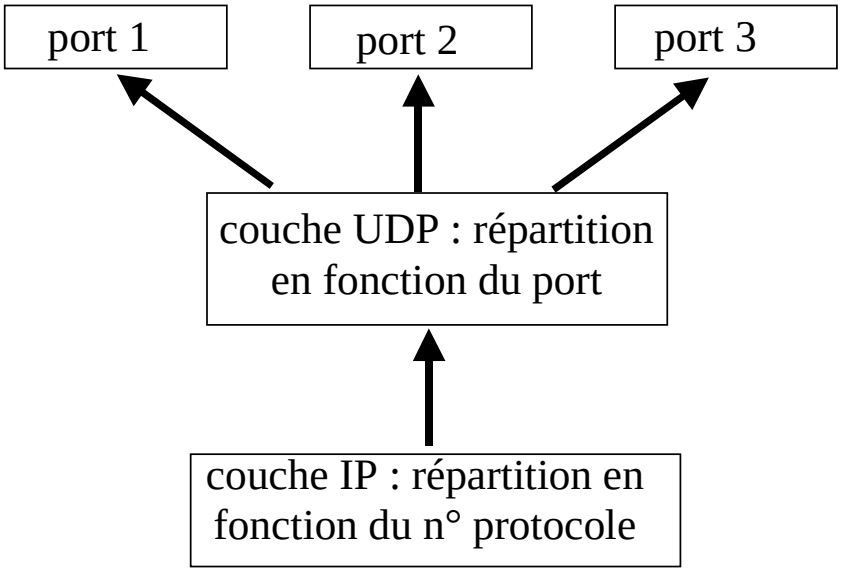
une destination est identifiée par un numéro de port (entier positif sur 16 bits)

les communications aux moyen des ports sont :
- tamponnées
- synchrones

Encapsulation:



Répartition suivant le port :



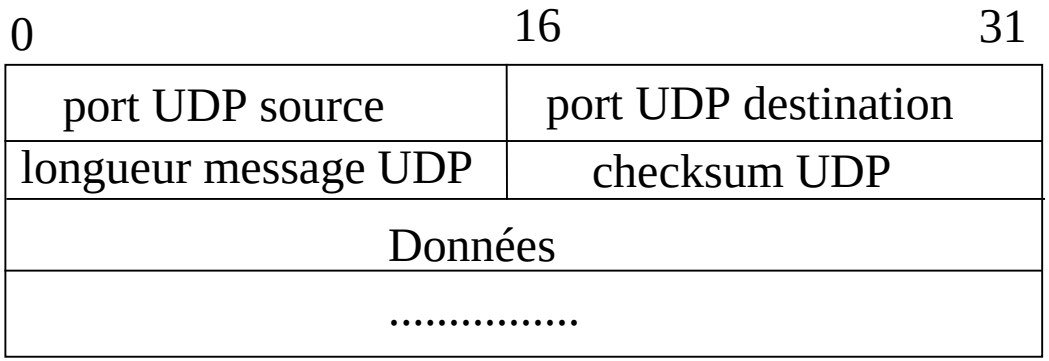
Affectation des numéros de ports :

un certain nombre de ports sont affectées définitivement
(well-known port assignments)
(disponibles sur les machines unix dans /etc/services)

les autres peuvent être utilisés par affectation dynamique
(il faut envoyer une requête à la machine cible pour connaître)
(localement on peut utiliser la fonction *getservbyname()*)

les ports sont spécifiques pour chaque protocole de transports

Format d'un message UDP:



Le port source est optionnel. doit contenir le numéro du port auquel il faut répondre et 0 sinon.

La longueur contient le nombre d'octets du message, en-tête plus données. 64 K max

Le checksum est optionnel, la valeur 0 indique qu'il n'a pas à être vérifié

Algorithme de calcul du checksum identique à celui d'IP

Protocole TCP (Transmission Control Protocol)

Fonctionnalités

- transfert fiabilisé d'un flux d'octets (la suite d'octets reçues est exactement celle qui a été envoyée)
- un flux d'octets est découpé en segments qui sont encapsulés dans des paquets IP ; la source et la destination ignorent comment est fait le découpage
- connection des entités communicantes par un circuit virtuel identifié par un couple d'extremités
- connection full duplex (dans les deux sens à la fois)

Fiabilisation d'un transfert sur un service non fiable

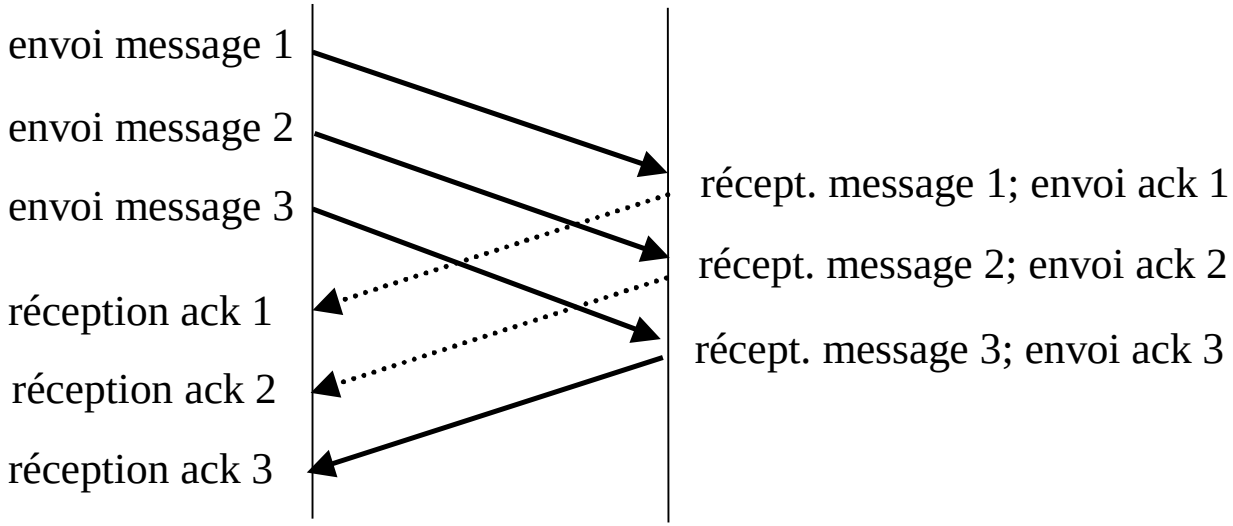
principe de base :

L'émission d'un segment s'accompagne de la mise en route d'un décompteur (timer).

Si le segment arrive à destination, le récepteur doit envoyer un accusé de réception

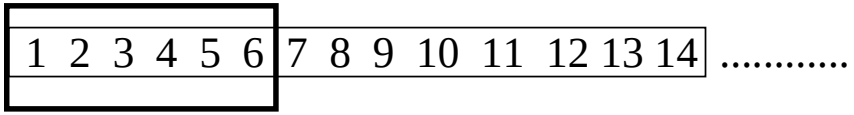
Si le décompteur arrive à 0 (time out) alors l'émetteur doit émettre à nouveau le segment.

Factorisation des accusés :

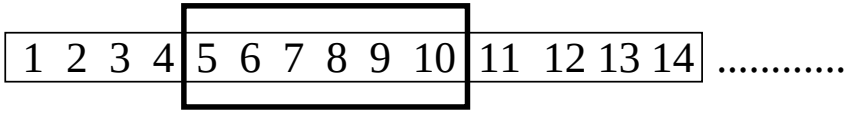


Fenêtre glissante :

fenêtre glissante de taille 6. Position initiale



après réception de ack 4



Mécanisme TCP

- numérotation des octets (et non des segments)

- le numéro d'octet dans un ack donne le numéro du prochain octet attendu (accuse réception de tous les octets précédents dans le flux)

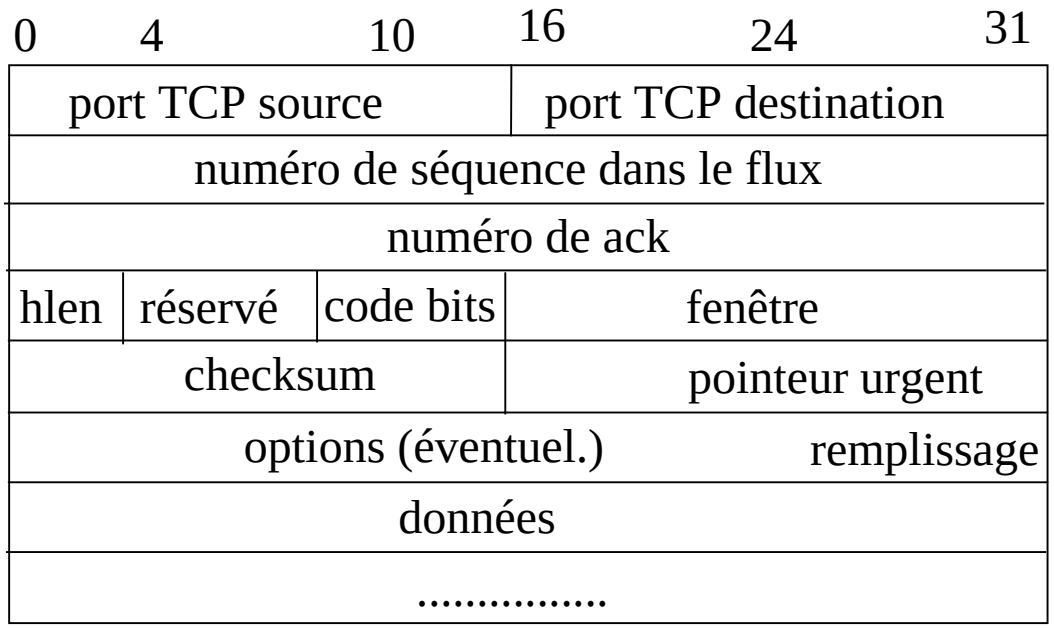
- fenêtre de taille dynamique (un ack peut augmenter ou réduire la taille de la fenêtre en fonction des possiblités du récepteur)

- ack transportés dans les message du flux inverse (piggy backing)

- algorithme de retransmission dynamique avec un time out dynamique fonction du RTT (Round Trip Time)

- lorsque le RTT augmente fortement (indice de congestion dans le réseau) la taille de la fenêtre d'émission est divisée par 2 et ainsi de suite

Format d'un segment TCP



- la connexion est identifiée par le couple (port local, port distant)

- plusieurs connections peuvent partager un port

code bits (de gauche à droite) :

URG le pointeur urgent est valide
ACK le champ ack est valide
PSH le segment demande un push
(délivrance de tous les octets envoyés)
RST la connection est rompue
SYN initialisation de no de séquence
FIN fin du flux de données

Données urgentes

- le bit URG indique que le segment contient des données urgentes jusqu'au pointeur urgent.
- les données urgentes doivent être délivrées au plus tôt à l'application destinataire.

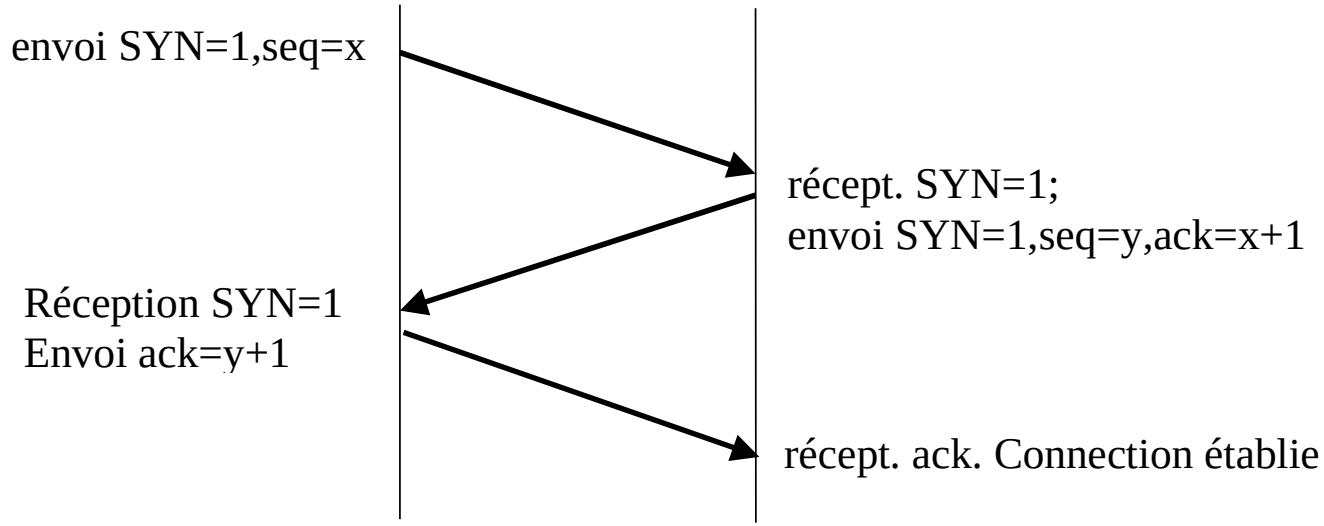
Ouverture d'une connection TCP

Une des deux application fait une ouverture passive :

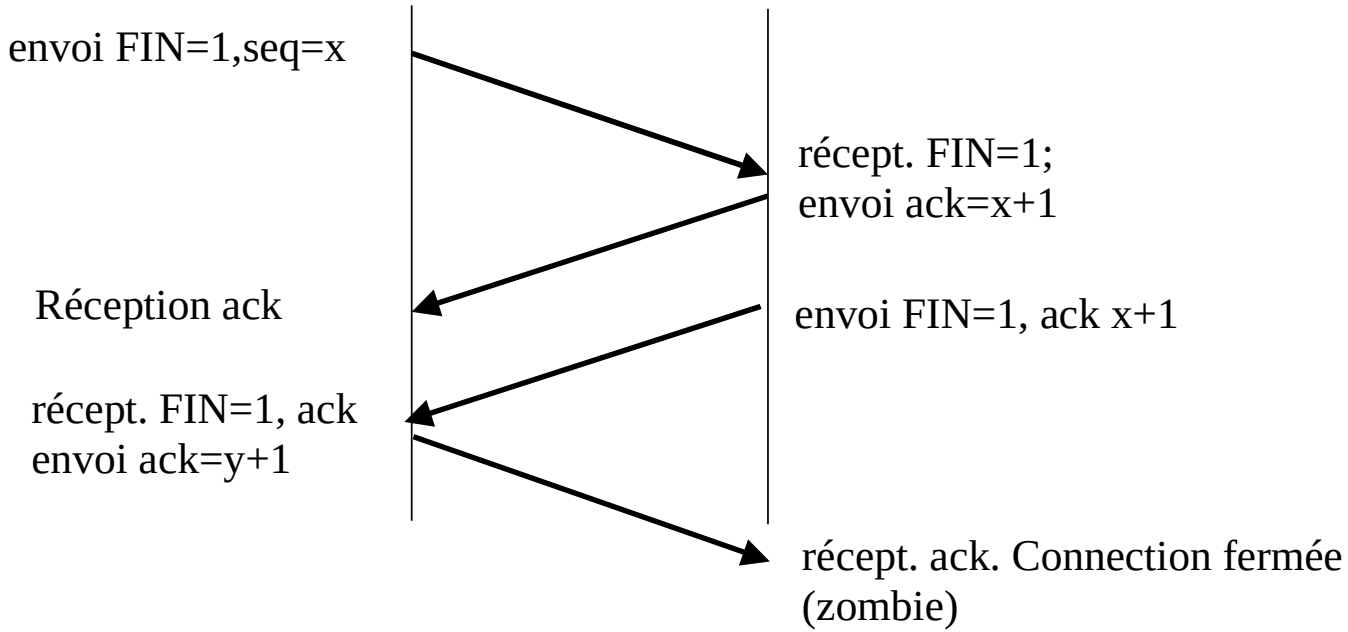
Elle indique qu'elle est prête à recevoir une demande de connection

L'autre peut alors faire une ouverture active pour établir la connection

Procédure en trois temps (three-way handshake)



Fermeture d'une connection



Représentation interne des données en machine et sur IP

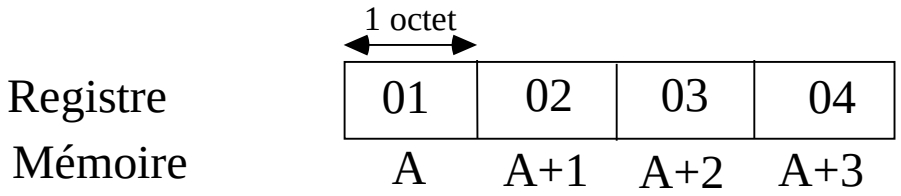
Most Significant Byte First (MSBF alias **Big Endian**) (Motorola...)

contre

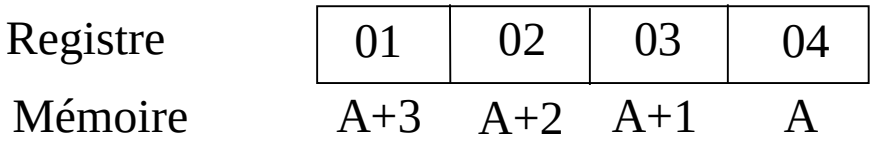
Least Significant Byte First (LSBF alias **Little Endian**) (Intel &co)

Exemple : nombre 0x01020304

Big Endian : on numérote les octets de gauche à droite



little endian : on numérote les octets de droite à gauche



Dans les paquet IP tous les champs sont en format **Big Endian.**

==>

Sur une machine les champs des structures de données sont en Big Endian

Les différents formats utilisés :

- Notation pointée (dotted format) « a » pour "ascii"
- Notation réseau (long) « n » pour "net", en big endian
- Représentation interne (dépendant de la machine) « h » pour "host"

- Format « Unix » d'une adresse réseau Internet « n »:

  #include <sys/type.h>
  #include <netinet.h>

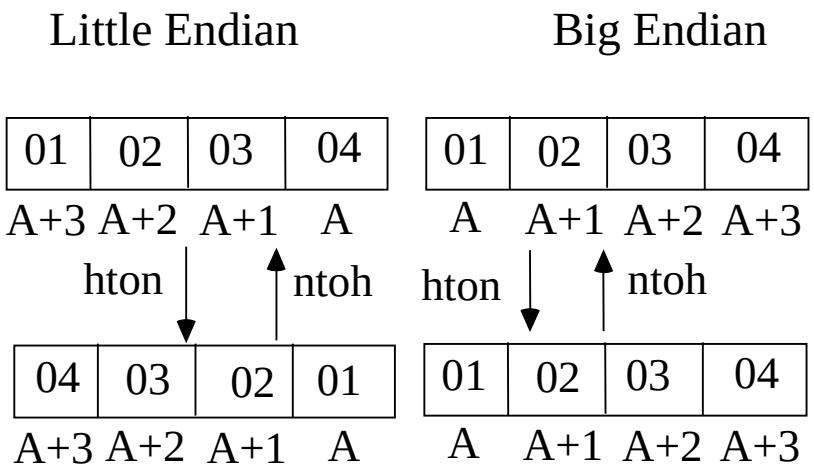
  struct in_addr {ulong s_addr}; entier long de 32 bits

Remarque : pour de nombreuses fonctions qui demandent un paramêtre en notation pointée, on peut, si la machine à accès à un service de noms (NIS ou Domain Name Service), utiliser à la place :

- Nom symbolique ou externe (exemple ulysse.iie.cnam.fr)

Conversion entre Big Endian et Little Endian

- Conversion entre format réseau « n » et format machine « h » : hton(), ntoh()
- Conversion entre format réseau « n » et format pointé « a » : ntoa(), aton()



Format Réseau

- Les même fonctions existent en entier court, suffixe s, et en entier long, suffixe l

Obtenir l'adresse au format « n » d'une station

La fonction de conversion *gethostbyname* permettent de retrouver une adresse IP au format « n » connaissant l'adresse en notation pointée ou le "nom" symbolique de la machine.

- *struct hostent *gethostbyname (char *hostname)*

Retourne une structure *hostent*, décrite dans *netdb.h*

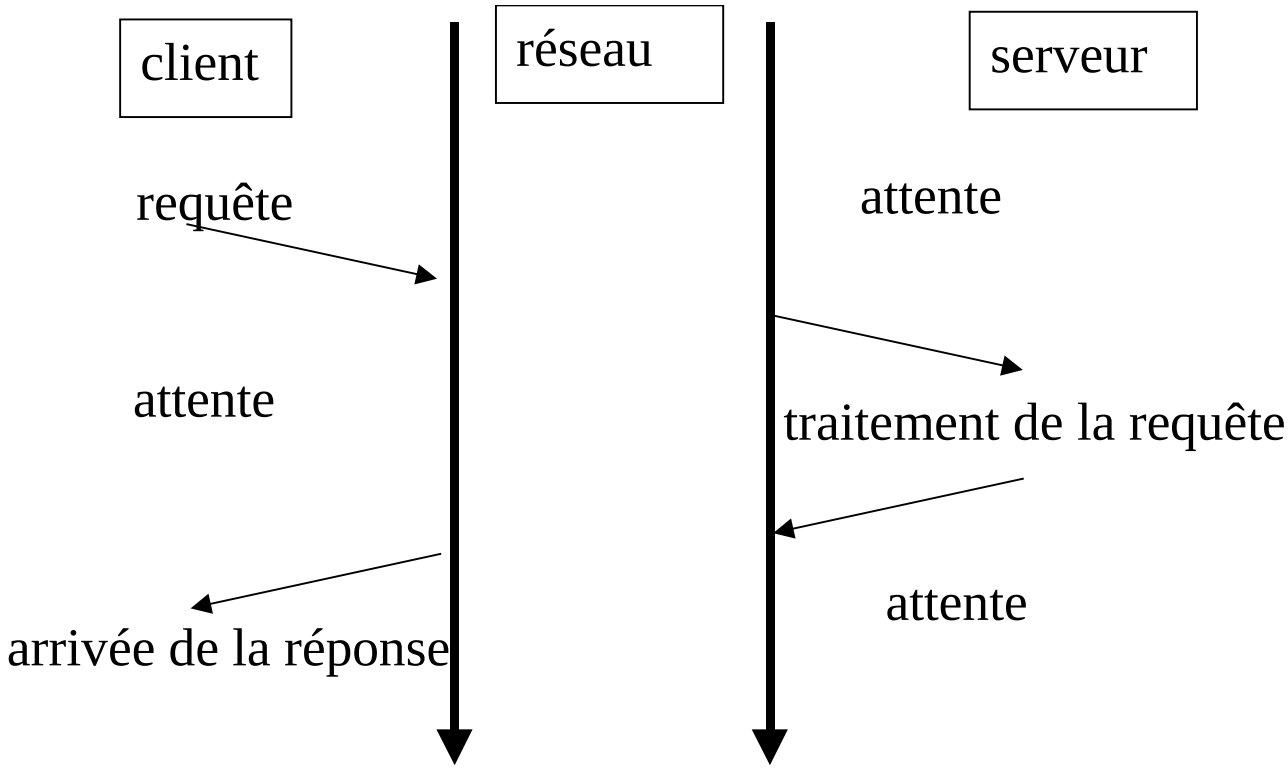
```
#include <netdb.h>

struct hostent{
    char    *h_name
    char    **h_aliases
    int     h_addrtype/*AF_INET*/
    int     h_length
    char    **h_addr_list
    };
```

Le dernier champ est une liste, terminée par NULL, d’adresses IP au format *in_addr*.

Il exist une fonction duale *gethostbyadress*

Schema Serveur/Clients



- Le serveur doit exister et être en attente avant que la requête arrive; le client a l'initiative du dialogue

Serveur avec ou sans états

- Pour avoir un dialogue suivi avec un client
- Pour remédier à l'effet de pertes de messages
  - un serveur peut conserver des informations sur un client
    - identification du client
    - degré d'avancement du dialogue avec le client
    - ....

    ==> Serveur avec état ou avec connexion
- Sinon serveur sans état (requête autonome)

Comment servir les requêtes ?

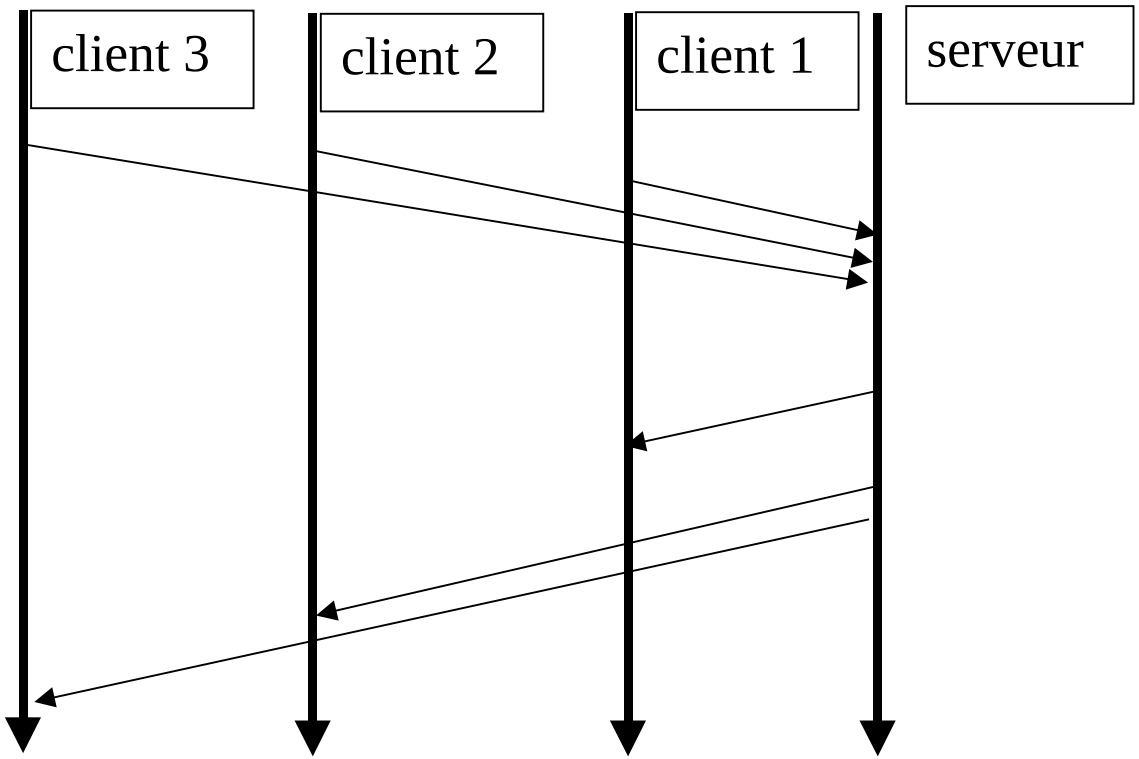
- Un client envoie une requête pour transferer 200 KO
- Un second client envoie une requête au même serveur pour transférer 1KO

- Peux-t-on attendre d'avoir fini de servir le premier client pour servir le second ?

=> Serveur séquentiel (itératif)

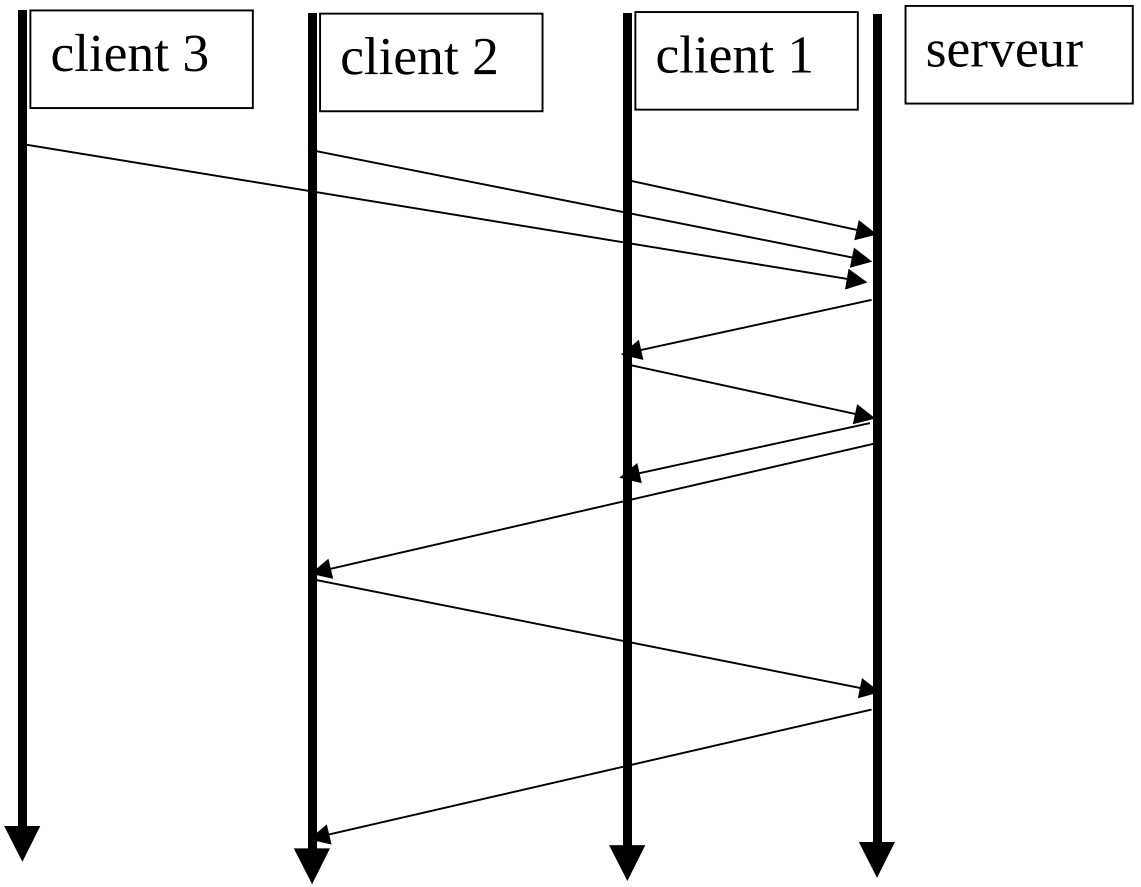
=> Serveur parallèle (concurrent)

Serveur séquentiel et réponses longues



- selon les requêtes en cours, le client peut attendre longtemps, même pour une réponse courte.

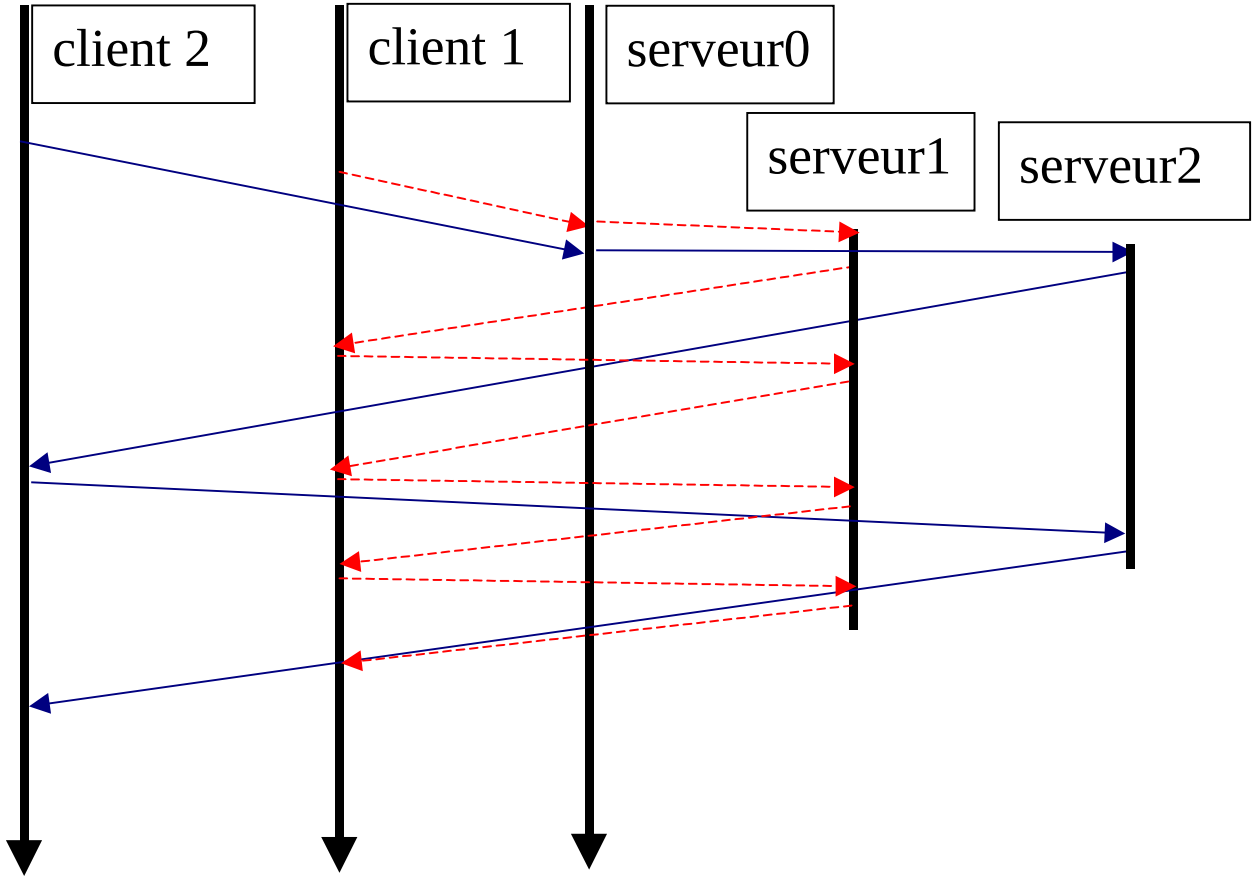
Serveur séquentiel en mode connecté



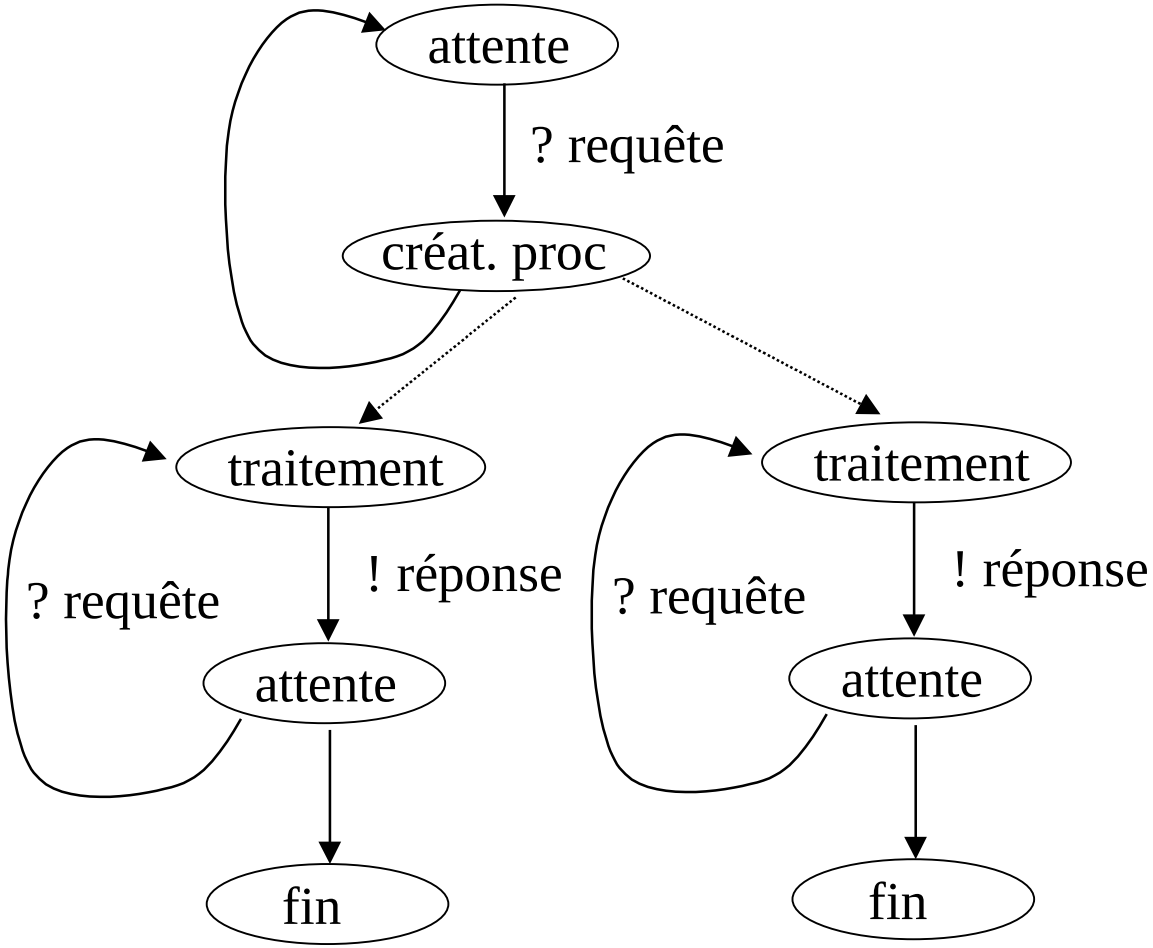
- selon la longueur des dialogues, le client peut attendre longtemps, même pour une réponse courte.

Serveur parallèle (multiprocessus ou multithreads)

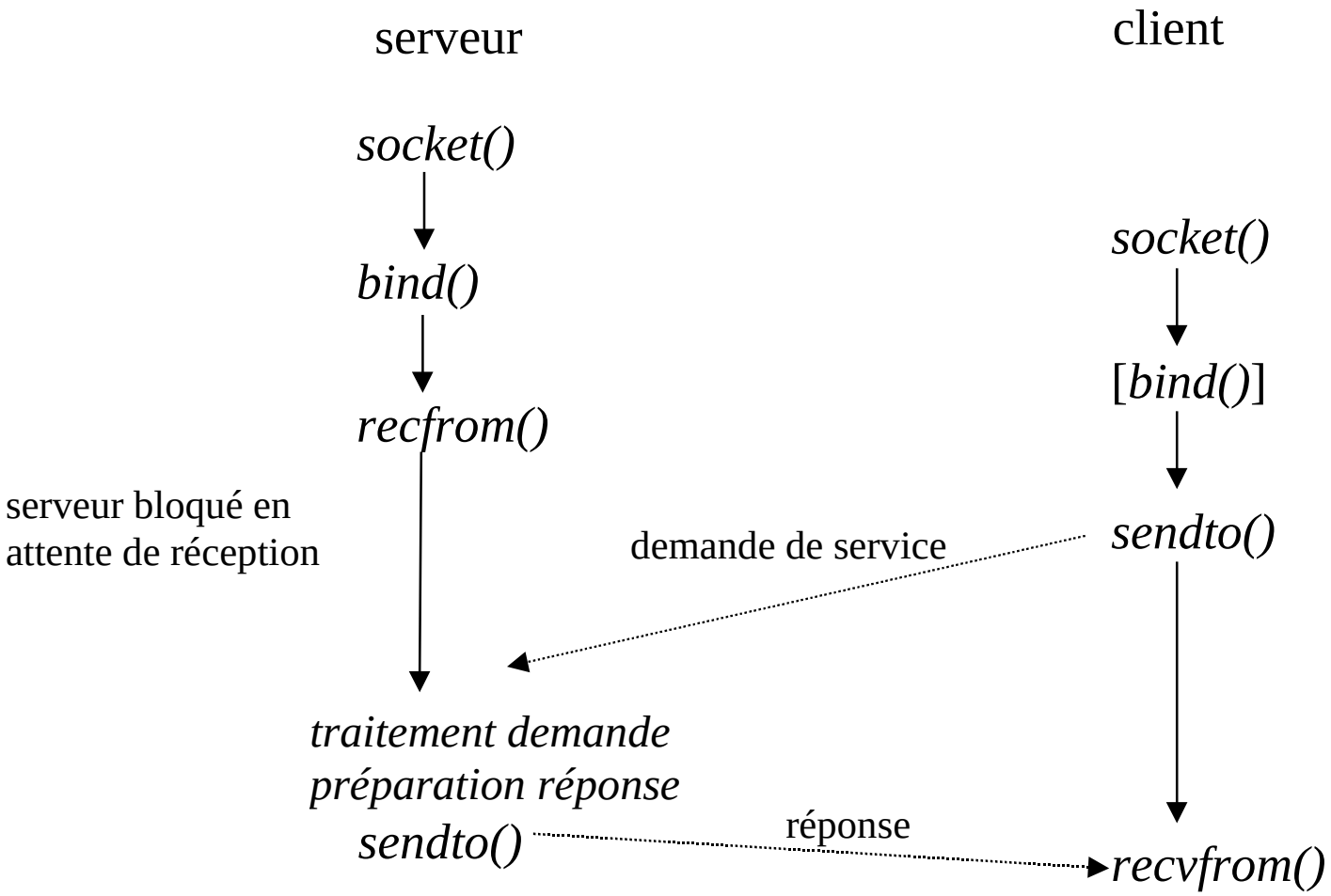
La gestion de plusieurs dialogues simultanés complique beaucoup la réalisation d'un serveur => traitement parallèles



Automate du serveur paralléle



Client/serveur mode non connecté: socket UDP



Client serveur en mode connecté : socket TCP