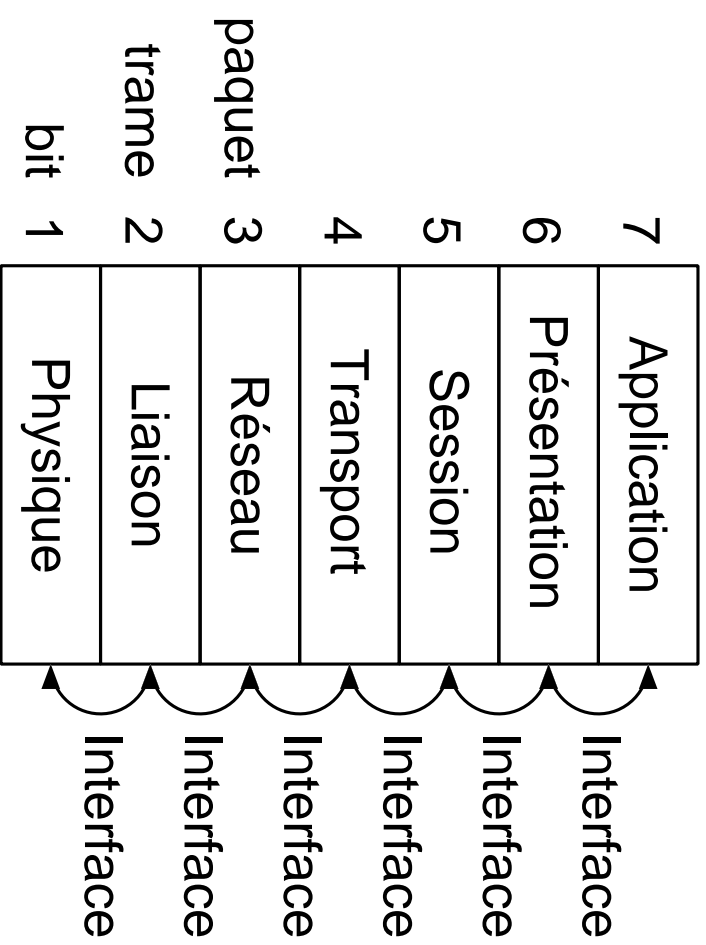
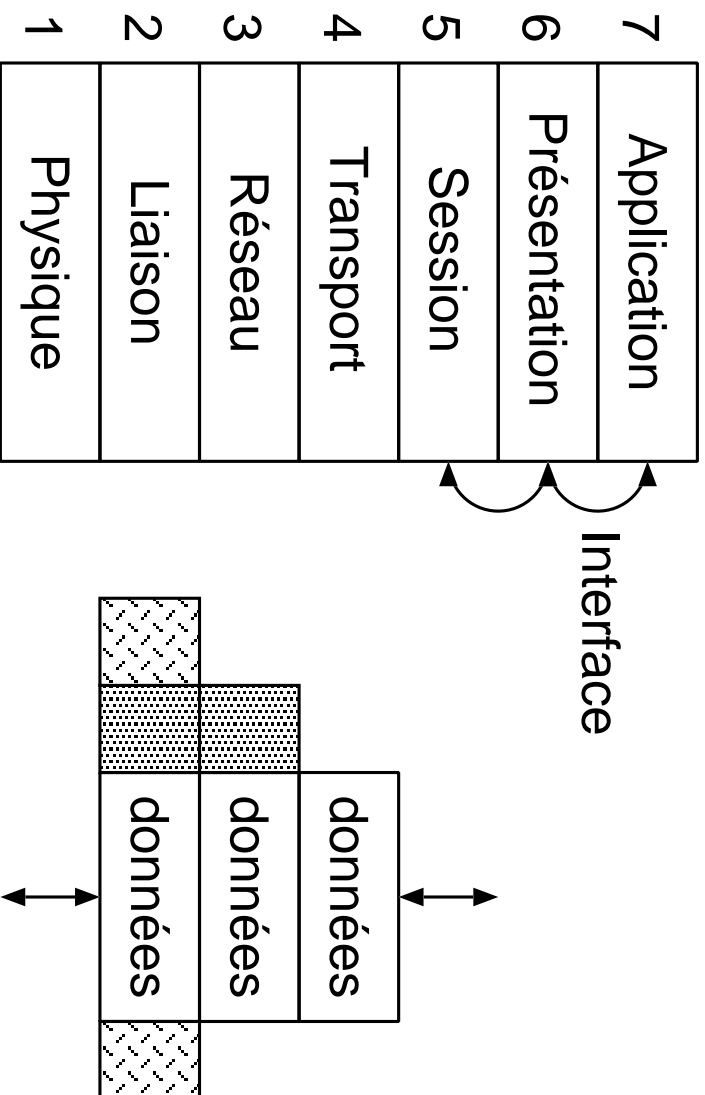


Les sept couches

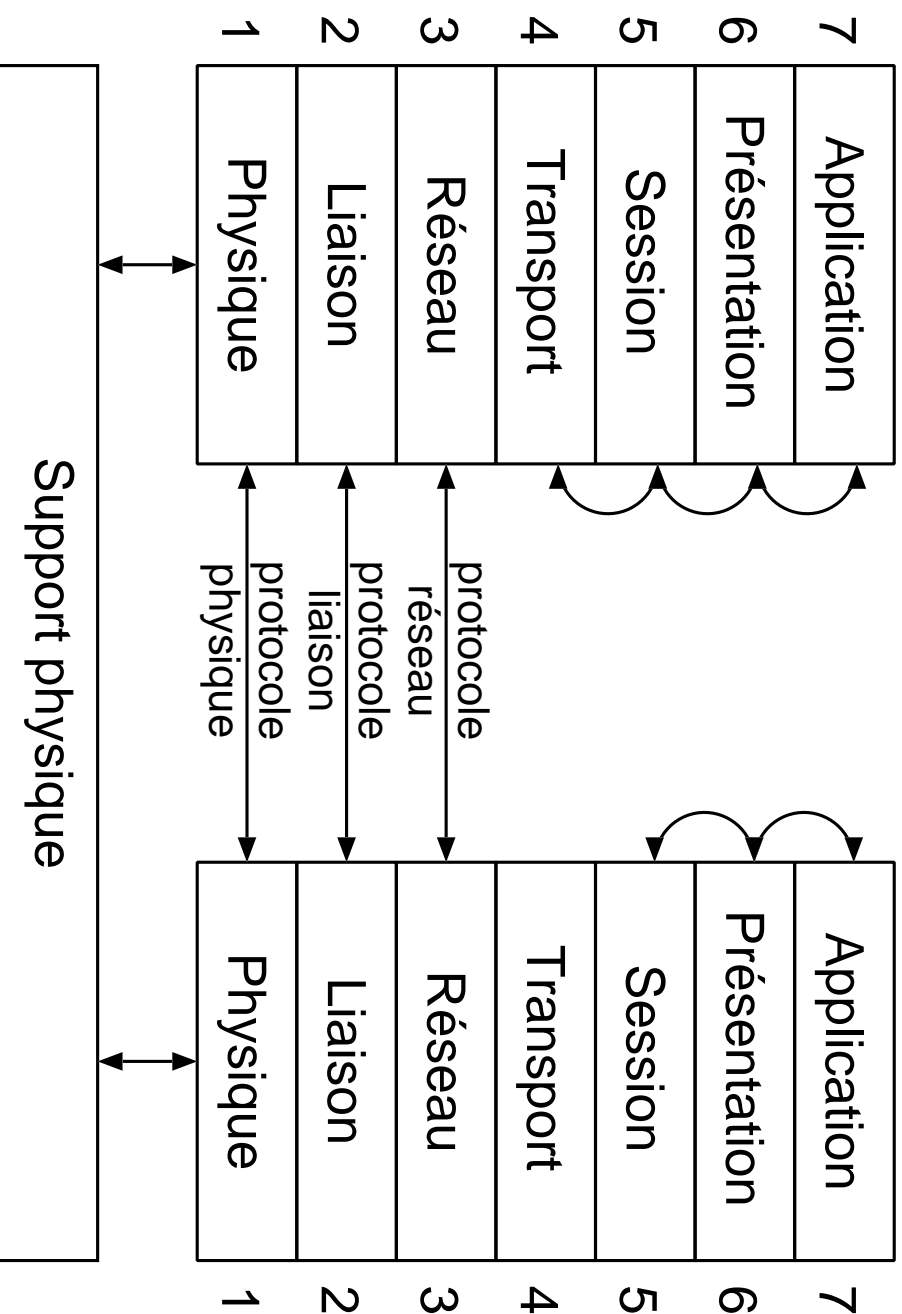


Transparent 1

Communication entre couches

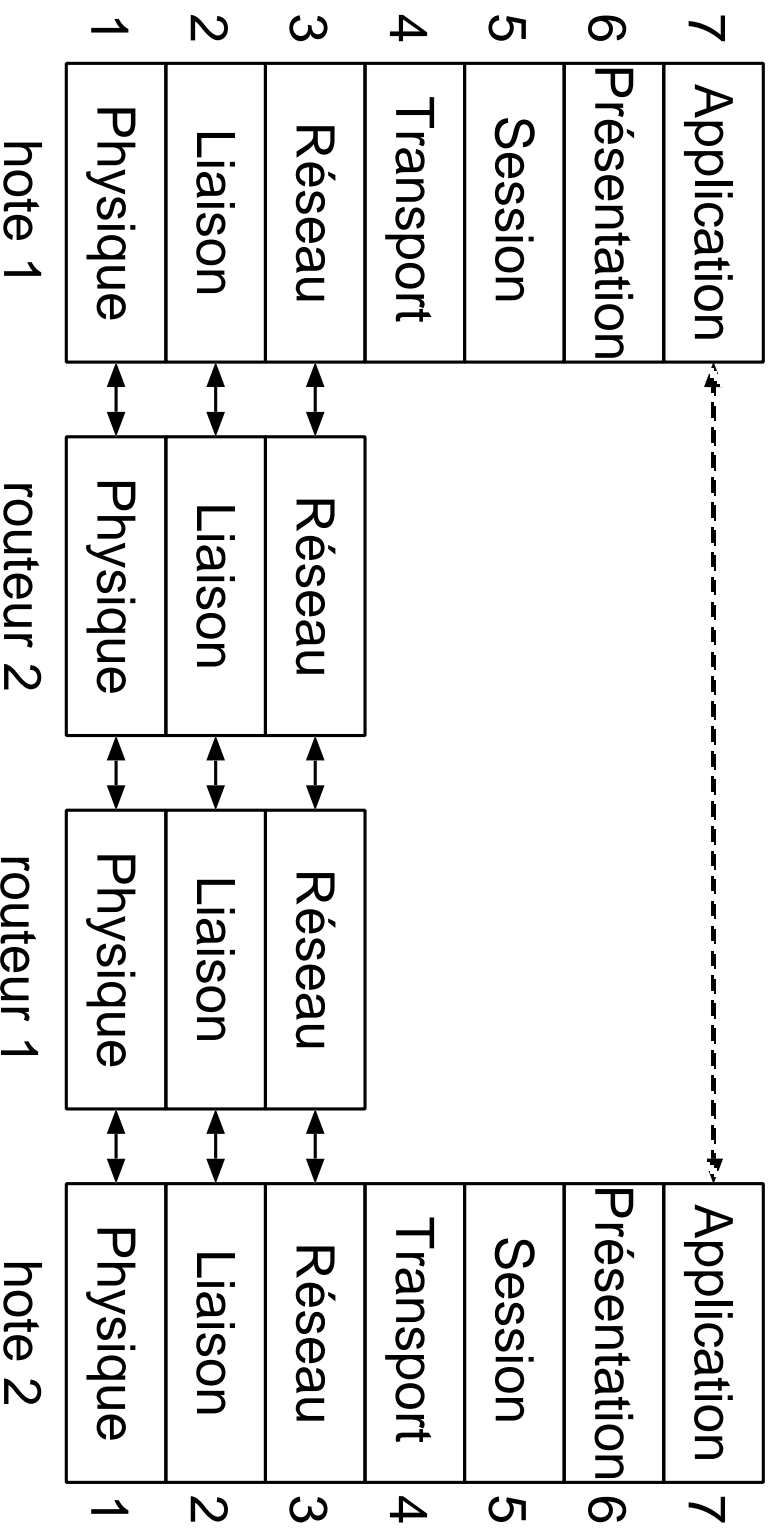


Communication entre deux systèmes



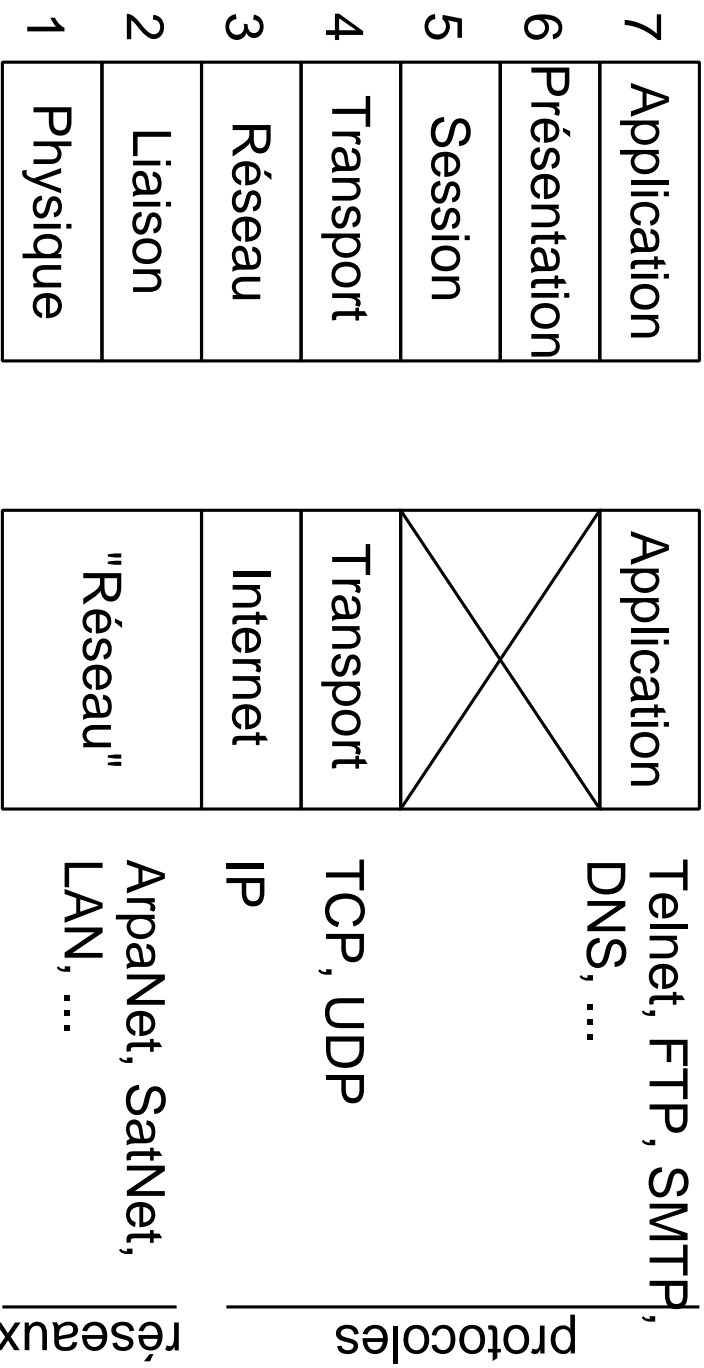
Transparent 3

Routeage



Transparent 4

Le modèle TCP/IP



Et protocoles de contrôle : ICMP, ARP, RARP, BOOTP.

Adresse IP

Une adresse = un entier codé sur 32 bits (donc 4 octets)

Notation «décimale pointée» : A . B . C . D

Exemples :

- 193 . 104 . 111 . 65, en hexadécimal : C1686F41
- 10 . 0 . 0 . 2, en binaire : 0000101000000000000000000000010
- 157 . 159 . 40 . 55, en décimal 2644453431
- 123 . 343 . 231 . 665 n'est pas une adresse IP

Découpage en réseaux

Les adresses sont regroupées dans des **réseaux**.

Exemples :

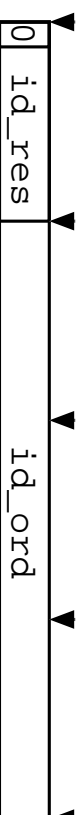
- l'ensemble des adresses commençant par 193.104.111
- celles commençant par 157.159.
- celles commençant par 34

On a donc une **arborescence** : de gauche à droite = du général au particulier (comme les n° de téléphone)

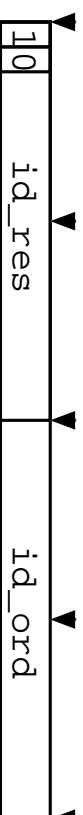
Classes IP

Des classes d'adresses définissent la grandeur des réseaux.

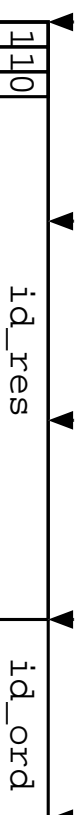
- **Classe A** : 126 réseaux de 16 millions d'adresses^a



- **Classe B** : 16 382 réseaux de 65 536 adresses



- **Classe C** : 2 millions de réseaux de 254 adresses



^aid_res : identifiant réseau et id_ord : identifiant ordinateur

Adresses spéciales

Deux adresses spécifiques définissent le réseau :

Adresse de réseau : donnée par la valeur «tout à 0»

Exemples : 157.159.0.0, 193.104.111.0

Adresse de diffusion (*broadcast*) : donnée par la valeur «tout à 1»

Exemples : 157.159.255.255, 193.104.111.255

Adresses spéciales «non diffusables»

127.*.*.* : adresses de rebouclage (*loopback*)

0.0.0.0 et 0.id_ord : «cet» ordinateur sur «ce» réseau (au démarrage)

ifconfig

aide en ligne : `man ifconfig`

Syntaxe **Linux** :

```
ifconfig interface options  
ifconfig interface <adresse> netmask xxx broadcast  
xxx
```

Exemple :

```
ifconfig eth0 193.104.111.122 netmask  
255.255.255.192 broadcast 193.104.111.127  
ifconfig eth0 up ou ifconfig eth0 down  
ifconfig sans argument donne la configuration actuelle.
```

Par rapport aux classes A,B,C

On va indiquer en plus de l'adresse une autre information : le **masque de réseau** ou *netmask*. C'est un nombre à 32 bits qui se compose d'un ensemble de 1 suivi d'un ensemble de 0.

- Les **1** indiquent la partie réseau
- Les **0** indiquent la partie machine

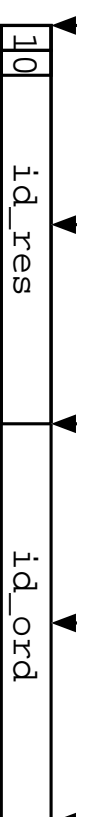
Exemples avec nos classes A, B et C :

- A : 11111111000000000000000000000000 soit 255.0.0.0
- B : 255.255.0.0
- C : le célèbre 255.255.255.0

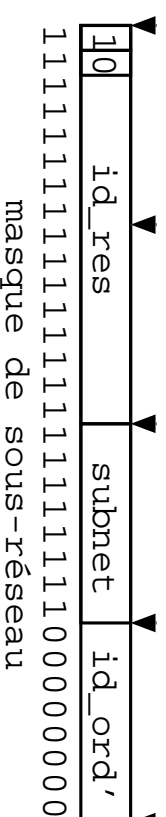
Exemple de découpage d'une classe B

Pour prendre un exemple simple, imaginons une classe B. Nous pouvons la découper en un ensemble de classe C en indiquant un masque adéquat.

Réseau vu de l'extérieur :



En interne, découpage en pseudo-classes C :



Note : ce redécoupage est invisible de l'extérieur de la classe B. Il est uniquement destiné à une gestion plus intelligente **en interne**.

Exemple de découpage d'une classe C

Réseau de base : 193 . 104 . 111 . 0 (classe C)

Découpage en 4 réseaux, masque à utiliser : 111 . . . 111 . 11000000 =
255 . 255 . 255 . 192

réseau	diffusion
193 . 104 . 111 . 0	193 . 104 . 111 . 63
193 . 104 . 111 . 64	193 . 104 . 111 . 127
193 . 104 . 111 . 128	193 . 104 . 111 . 191
193 . 104 . 111 . 192	193 . 104 . 111 . 255

4 réseaux de $256/4 - 2 = 62$ machines possibles

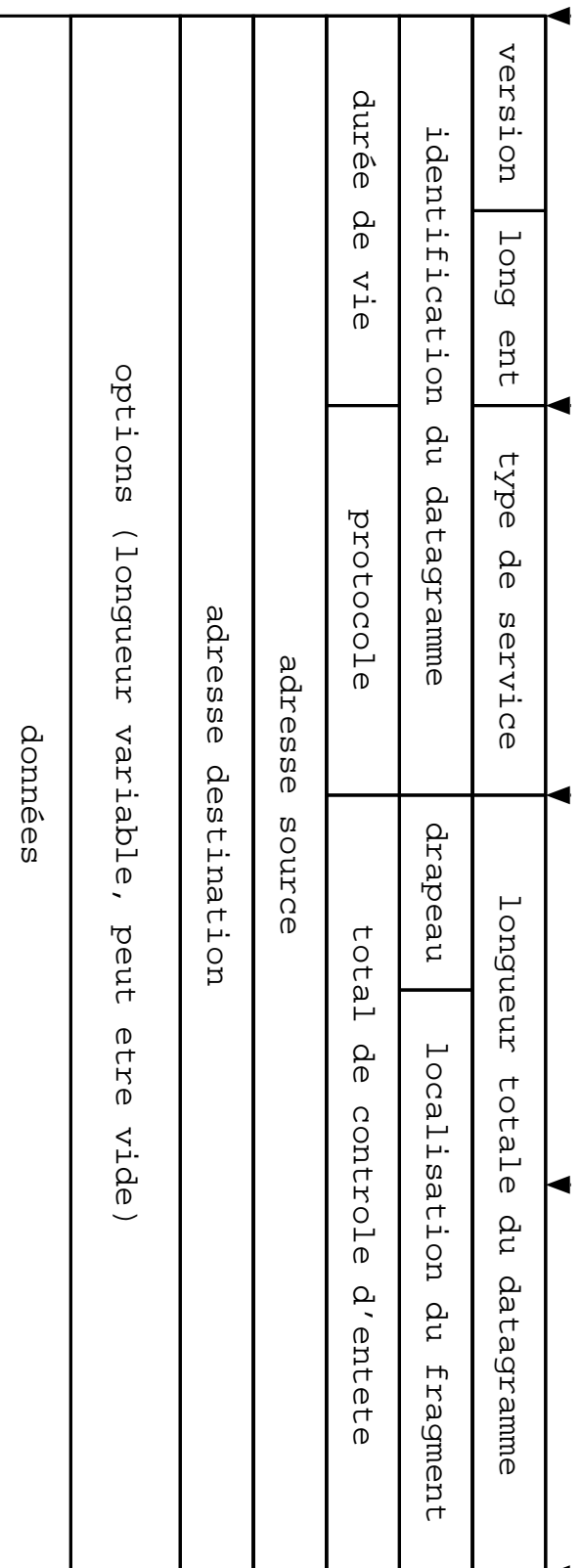
Autre notation

Puisque les masques ne représentent qu'un certain nombre de 1 complétés par des 0 (pour obtenir 32 bits), la seule information intéressante est ce nombre de 1. Ce nombre correspond au **nombre de bits bloqués**.

Quelques exemples :

- 10.0.0.0 masque 255.0.0.0 : **10.0.0.0/16**
- 207.50.2.0 masque 255.255.255.0 : **207.50.2.0/24**
- 202.86.0.0 masque 255.255.248.0 : **202.86.0.0/21**
- 202.86.0.0 masque 255.255.248.0 : **202.86.0.0/21**
- réseau de 8192 adresses : **202.86.16.0/20**
- réseau de 1024 adresses : **202.86.8.0/22**

Un datagramme IP

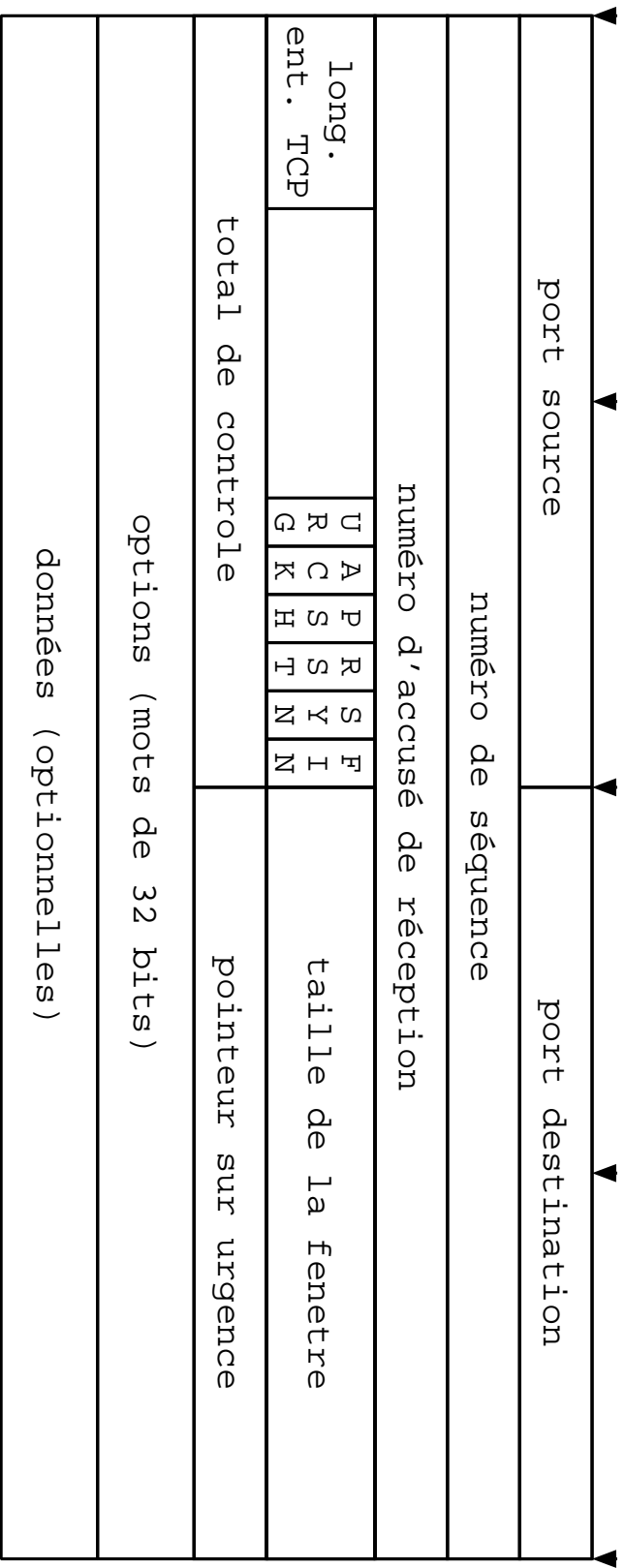


version : on est encore aujourd'hui en version 4

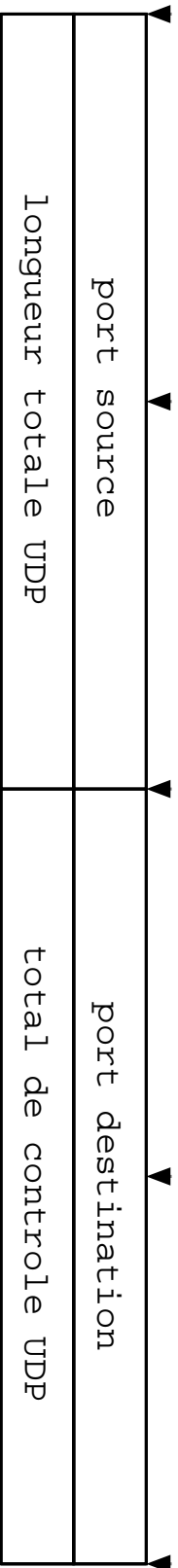
long ent : longueur de l'entête (* 32 bits), options comprises
 protocole : TCP, UDP, ICMP, etc.

options : extensions possibles pour le protocole

Entête TCP



Entête UDP



Ports et sockets

Port : numéro (nombre sur 16 bits) identifiant une connexion. Permet le **multiplexage** des connexions sur une machine. Derrière chaque port utilisé, on trouve une application qui communique ou est en attente de connexion.

Socket : couple **adresse** + **port** identifiant une connexion sur une machine.

Une connexion entre deux machines en TCP ou UDP est donc complètement définie par un **couple sockets** : la connexion sur le «serveur» et celle sur le «client».

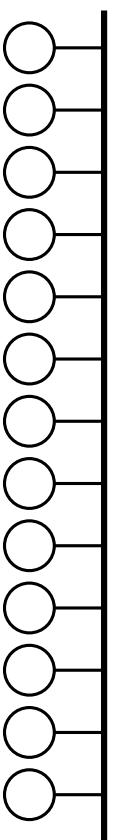
Pour voir les sockets «activées» sur une machine : `netstat -a`

Primitives de programmation TCP

client/serveur	primitive	signification
c & s	socket	création d'un point de connexion
s	bind	attachement de l'adresse au socket
s	listen	acceptation de connexion
s	accept	attente d'une connexion
c	connect	demande de connexion
c & s	send	envoi de données
c & s	receive	reception de données
c & s	close	fin de la connexion

TP 1 : un seul réseau

masques et adresses réseaux identiques



adresse : _____

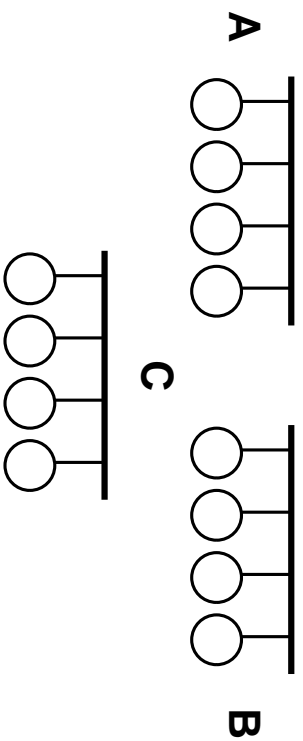
masque : _____

réseau : _____

diffusion : _____

notation CIDR : _____ / _____

TP 2 : trois réseaux



réseau : (A) (B) (C)

adresse : _____

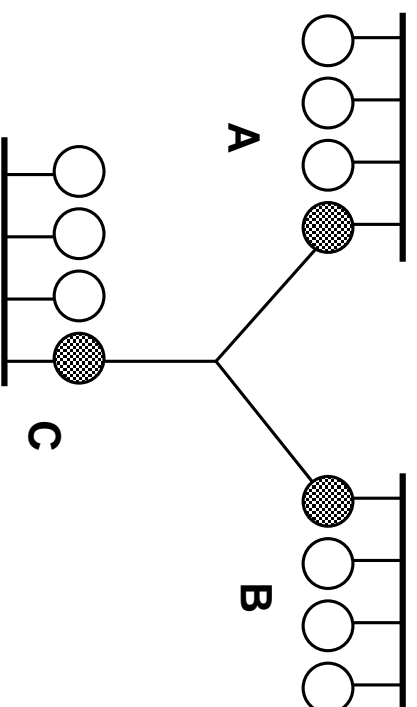
masque : _____

adr. réseau : _____

diffusion : _____

notation CIDR : _____ / _____

TP 3 : interconnexion



```
echo 1 > /proc/sys/net/ipv4/ip_forward
```