

Programmation Réseau



Jean-Baptiste.Yunes@univ-paris-diderot.fr

UFR Informatique

2012-2013

Introduction

- Ce cours n'est pas un cours de réseau
 - on y détaillera pas de protocoles de bas-niveau (mais on en parlera)
- Ce cours est un cours de **programmation** réseau
 - on y apprend à écrire des applications nécessitant de la **communication**

- Pour le programmeur, qu'est-ce qu'un réseau ?
 - un ensemble de services
 - un moyen de communication (ensemble de machines/applications atteignables)
 - ...

- Il existe au moins deux grands modes de communication :
- par **paquet** (datagram) : la poste en est un bon exemple
 - dans ce mode on est pas connecté (encore que), on est juste atteignable
- en **flux** (stream) : la téléphonie en est un bon exemple
 - dans ce mode on est connecté



- dans le mode **paquet** :
- il n'existe pas d'ordre dans la délivrance des paquets
 - un paquet posté en premier peut arriver en dernier
- il n'existe pas non plus de fiabilité
 - un paquet envoyé peut être perdu
- intérêt : souple et léger...

- dans le mode **flux** :
 - les informations sont reçues dans l'ordre exact de leur émission
 - il n'y a pas de perte
- inconvénient : nécessite l'établissement d'une connexion et consomme des ressources

- Comment ceci fonctionne-t-il dans la réalité ?
 - le service de base est un mode paquet non fiable
 - la **perte** peut être compensée par des ré-émissions
 - la **dégradation** peut être compensée par des codages
- Comment passe-t'on d'un tel service à des services fiables ?
 - En superposant des couches logicielles agrémentant les couches inférieures de propriétés attendues

- le modèle de référence est le modèle **ISO/OSI** (**O**pen **S**ystem **I**nterconnection) en 7 couches

OSI (Open Source Interconnection) 7 Layer Model

Layer	Application/Example	Central Device/ Protocols	DOD4 Model

Couche	Protocoles
Application	FTP, HTTP, SMTP, POP, SSH, TELNET, IMAP...
Présentation	SSL, WEP, WPA, Kerberos
Session	Ports
Transport	TCP, UDP, SPX
Réseau	IPv4, IPv6, ARP, IPX
Liaison	802.11, WiFi, ATM, Ethernet, ISDN
Physique	Fibre, Câble, Radio...

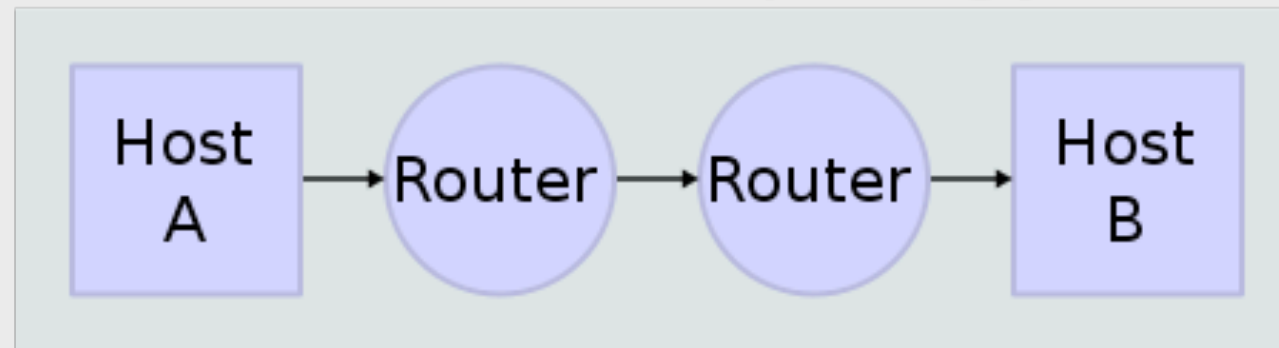
- Le modèle Internet est à 4 couches



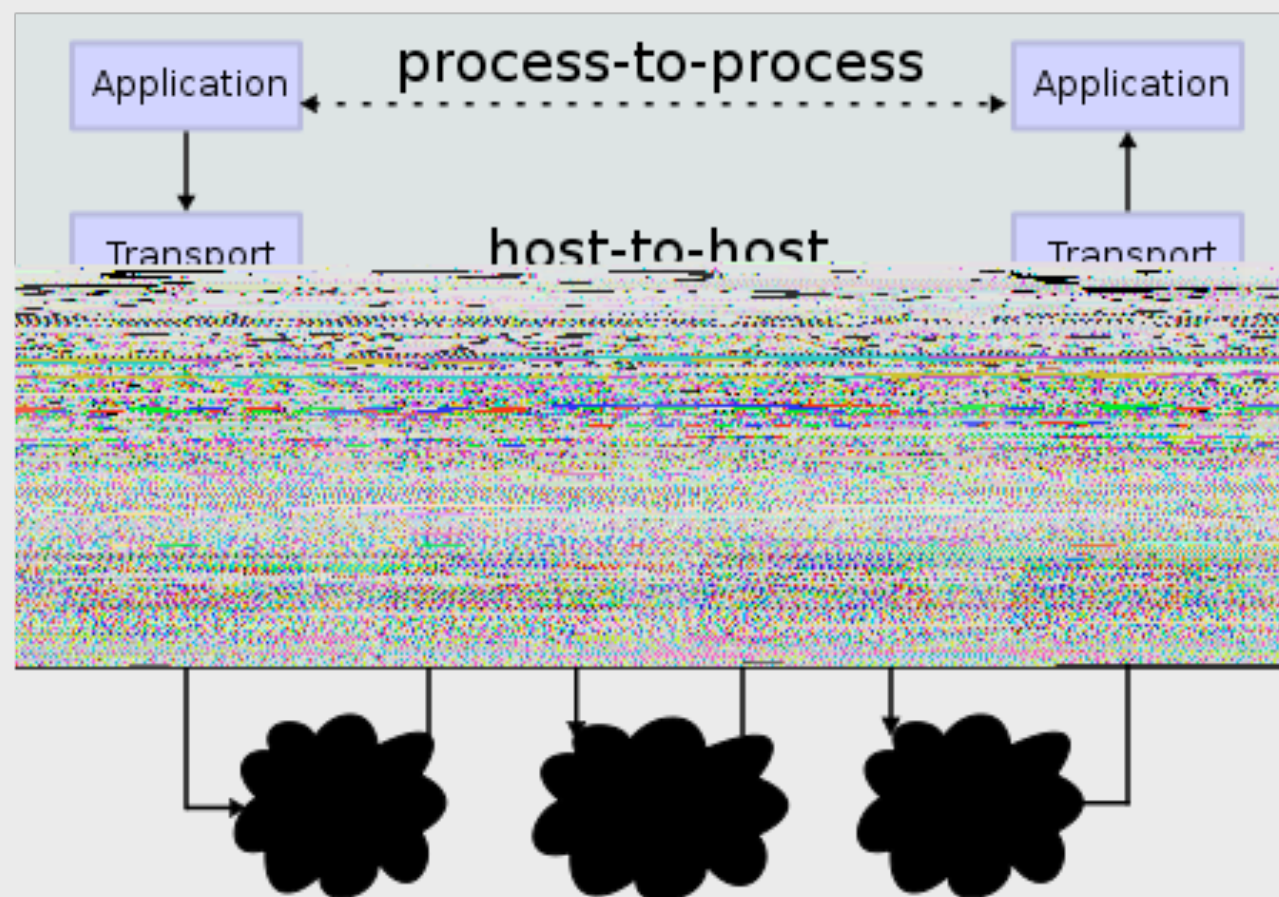
skokberg.eu

Couche	Protocoles
Application	FTP , HTTP , IMAP , POP...
Transport	TCP , UDP...
Internet	IPv4 , IPv6 , IPsec , ICMP...
Liaison	ARP , PPP , DSL , Ethernet...

Network Topology



Data Flow



Wikipedia

- Rappelons que dans le monde Internet la normalisation est disponible sous la forme de RFCs (**R**equest **F**or **C**omments)

- Les deux couches qui nous intéressent seront donc :
- la couche transport :
 - UDP (**U**ser **D**atagram **P**rotocol), RFC 768
 - TCP (**T**ransmission **C**ontrol **P**rotocol), RFC 793
- la couche application :
 - très variée...

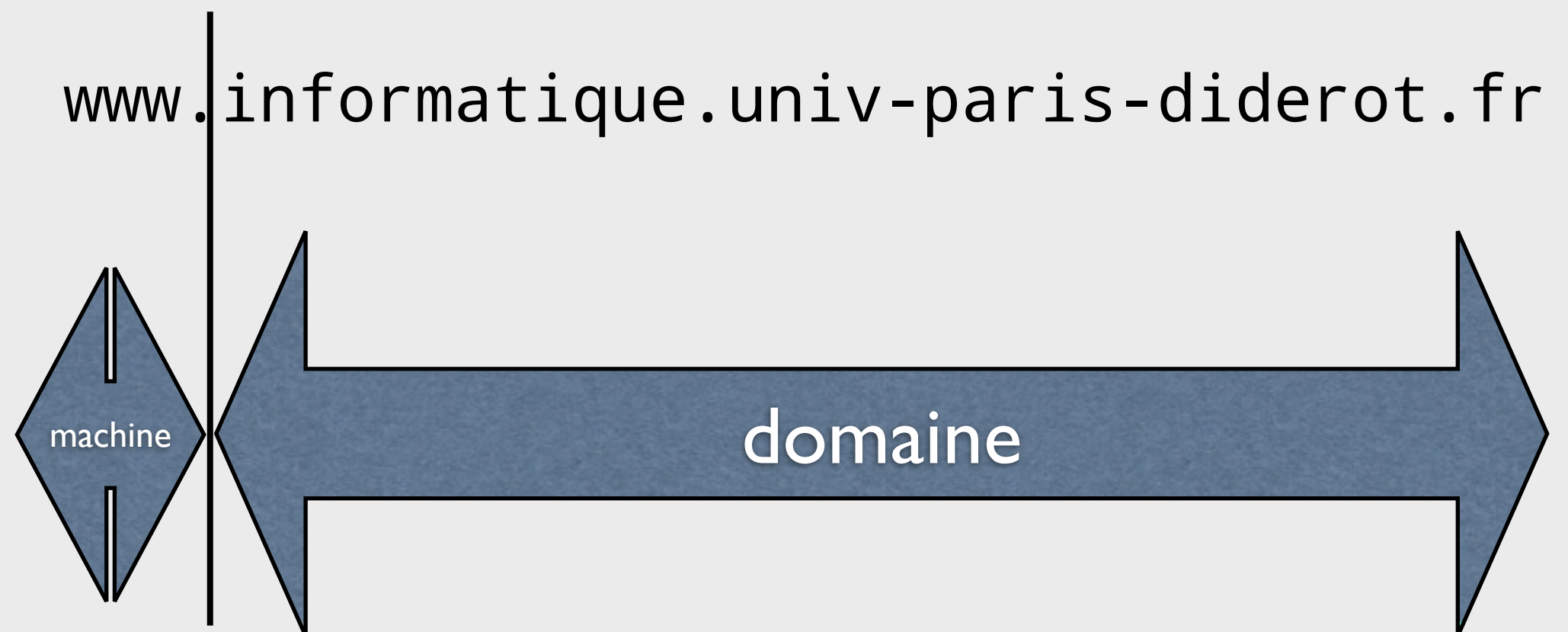
- Le couche transport fournit un ou plusieurs services de communication entre applications (par exemple TCP ou UDP)
 - pour communiquer, ces applications doivent se connaître
 - il existe donc un mécanisme de **nommage**...
- une application fournit un service particulier sur une machine donnée
 - le service est identifié par un **port**
 - la machine est identifiée par une **adresse**
 - ces informations sont similaires à celles jouées par un numéro de bureau (port) dans une administration donnée (machine)

- un couple (adresse,port) est un **point de communication**
- toute **communication** ne peut s'effectuer qu'entre au moins **deux points** de communications
 - l'émetteur
 - le (ou les) **receveur(s)**

- une machine peut-être identifiée par :
 - un **nom Internet** (pas strictement nécessaire)
 - par exemple : www.samsung.com
 - une machine peut posséder plusieurs noms
 - par une **adresse Internet** (toute machine connectée au réseau en possède une)
 - en réalité il s'agit de l'adresse d'un dispositif réseau sur une machine (ex : 211.45.27.202)
 - une adresse par dispositif
 - mais possiblement plusieurs dispositifs pour une machine

- Les adresses correspondent à une organisation structurelle du réseau
- Les machines préfèrent les nombres
- Les noms correspondent à une organisation logique
- Les humains préfèrent les noms
- Pensez à la situation des liens et des inœuds dans les systèmes de fichiers de la famille Unix... inœud → structure disque, lien → structure logique

- Le nom entièrement qualifié d'une machine est une représentation hiérarchique de la structure logique à laquelle elle appartient
- Le nom est constitué de deux parties



- La spécification du domaine représente la hiérarchie des responsabilités
- exemple : l'ufr d'informatique de l'université paris diderot située dans le domaine français
- le domaine le plus à droite est appelé domaine de premier niveau (top-level domain)
 - en gros, deux types :
 - génériques (ex. : .com, .edu, ...)
 - nationaux (ex. : .fr, .tz, ...)

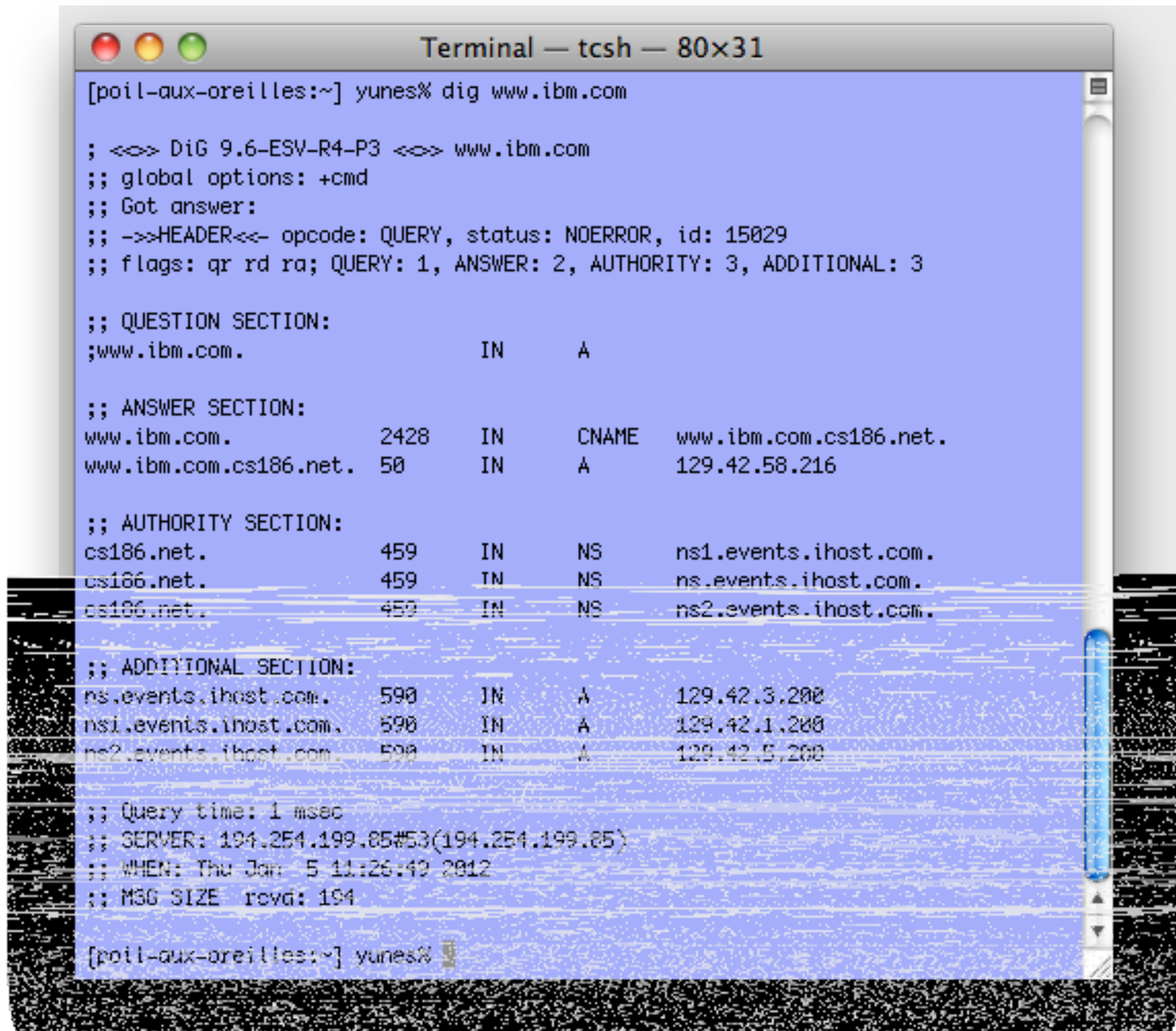
- pour `www.informatique.univ-paris-diderot.fr`
- `fr` est le domaine national attribué par l'ICANN à la france avec délégation à l'AFNIC
- `univ-paris-diderot` est le sous-domaine attribué par l'AFNIC à l'université paris diderot avec délégation à la DSI de l'université
- `informatique` est le sous-domaine attribué par la DSI à l'UFR avec délégation au service informatique de l'UFR
- `www` est le nom d'une des machines sous la responsabilité de l'UFR

- Les adresses aussi sont structurées
 - mais la structure est un reflet de la structure physique du réseau, du moins en théorie
 - dans ce cours la structure des adresses ne nous intéresse pas
 - cela relève du domaine pur des réseaux...

- Le service permettant de faire la translation d'un nom en une adresse s'appelle le **service de nom**, c'est un annuaire
- le système aujourd'hui le plus répandu est le DNS (RFC 882) qui est un service d'annuaire distribué
- Important : les communications nécessitent la connaissance des adresses Internet des machines concernées
- On notera au passage qu'il existe aujourd'hui deux types d'adresses : IPv4 et IPv6 (respectivement 4 et 8 octets)
 - deux réseaux qui cohabitent...

site
ROT

- le service d'annuaire distribué peut être interrogé à l'aide d'outils (des APIs sont disponibles comme on le verra plus tard)
- Outils :
 - dig
 - nslookup (souvent considéré comme obsolète)

A screenshot of a macOS Terminal window titled "Terminal — tcsh — 80x31". The window has a blue background and a white border with standard macOS window controls (red, yellow, green buttons). The terminal shows the output of a `dig www.ibm.com` command. The output is formatted in a structured way, showing the query details, question section, answer section, authority section, and additional section. The answer section shows that www.ibm.com is a CNAME record pointing to www.ibm.com.cs186.net, which is an A record with IP address 129.42.58.216. The authority section shows three NS records for cs186.net pointing to ns1.events.ihost.com, ns.events.ihost.com, and ns2.events.ihost.com. The additional section shows three A records for these nameservers. The query time is 1 msec, the server is 194.254.199.85, and the query was made on Thu Jan 5 11:26:49 2012.

```
[poil-aux-oreilles:~] yunes% dig www.ibm.com

; <<>> DiG 9.6-ESV-R4-P3 <<>> www.ibm.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15029
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 3

;; QUESTION SECTION:
;www.ibm.com.                IN      A

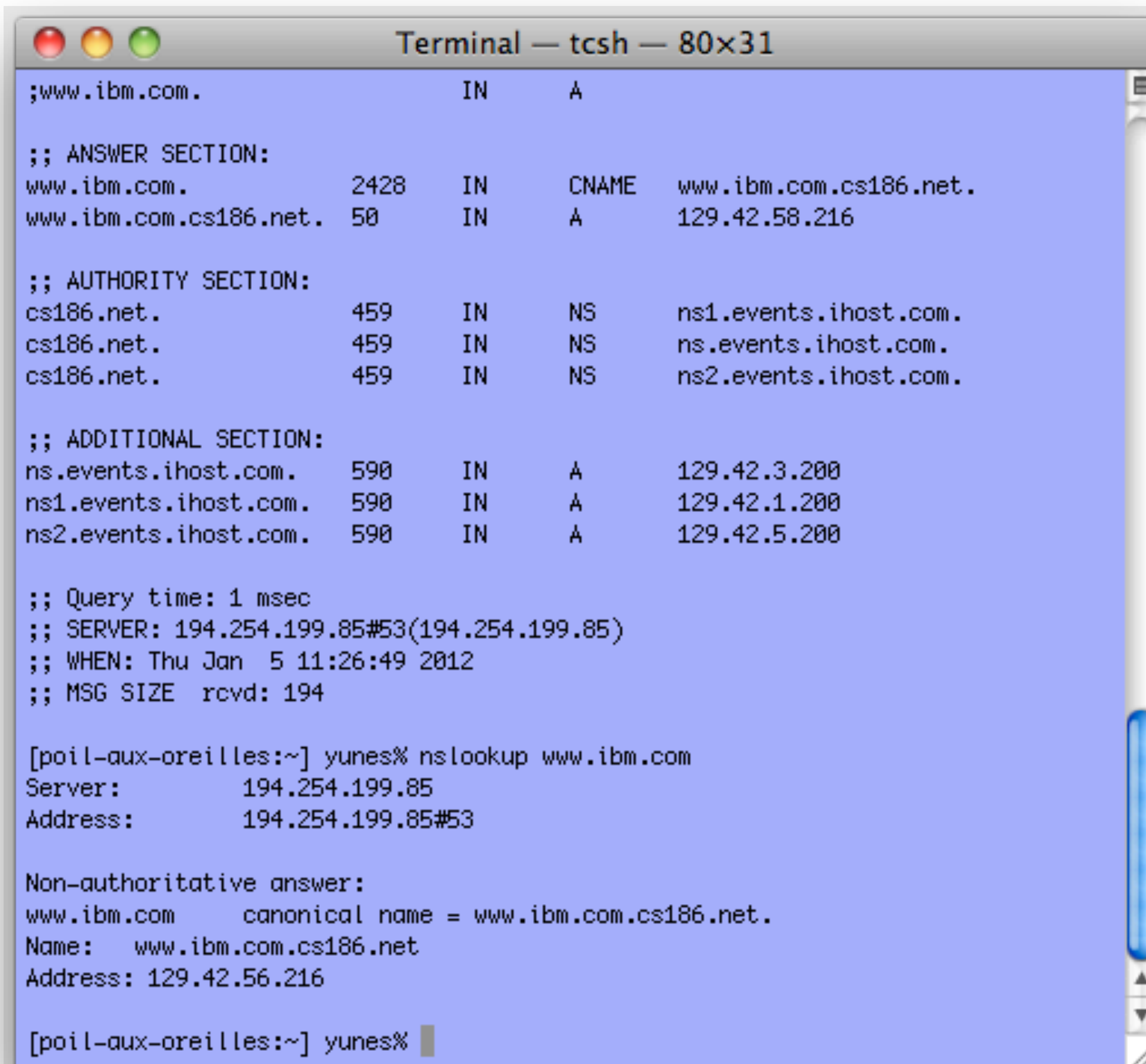
;; ANSWER SECTION:
www.ibm.com.                 2428    IN      CNAME   www.ibm.com.cs186.net.
www.ibm.com.cs186.net.       50      IN      A       129.42.58.216

;; AUTHORITY SECTION:
cs186.net.                   459     IN      NS      ns1.events.ihost.com.
cs186.net.                   459     IN      NS      ns.events.ihost.com.
cs186.net.                   459     IN      NS      ns2.events.ihost.com.

;; ADDITIONAL SECTION:
ns.events.ihost.com.         590     IN      A       129.42.3.200
ns1.events.ihost.com.        590     IN      A       129.42.1.200
ns2.events.ihost.com.        590     IN      A       129.42.5.200

;; Query time: 1 msec
;; SERVER: 194.254.199.85#53(194.254.199.85)
;; WHEN: Thu Jan  5 11:26:49 2012
;; MSG SIZE  rcvd: 194

[poil-aux-oreilles:~] yunes%
```



```
Terminal — tcsh — 80x31

;www.ibm.com.                IN      A

;; ANSWER SECTION:
www.ibm.com.                 2428    IN      CNAME   www.ibm.com.cs186.net.
www.ibm.com.cs186.net.       50      IN      A        129.42.58.216

;; AUTHORITY SECTION:
cs186.net.                   459     IN      NS       ns1.events.ihost.com.
cs186.net.                   459     IN      NS       ns.events.ihost.com.
cs186.net.                   459     IN      NS       ns2.events.ihost.com.


;; ADDITIONAL SECTION:
ns.events.ihost.com.         590     IN      A        129.42.3.200
ns1.events.ihost.com.        590     IN      A        129.42.1.200
ns2.events.ihost.com.        590     IN      A        129.42.5.200

;; Query time: 1 msec
;; SERVER: 194.254.199.85#53(194.254.199.85)
;; WHEN: Thu Jan  5 11:26:49 2012
;; MSG SIZE  rcvd: 194

[poil-aux-oreilles:~] yunes% nslookup www.ibm.com
Server:          194.254.199.85
Address:         194.254.199.85#53

Non-authoritative answer:
www.ibm.com      canonical name = www.ibm.com.cs186.net.
Name:   www.ibm.com.cs186.net
Address: 129.42.56.216

[poil-aux-oreilles:~] yunes%
```

A screenshot of a macOS Terminal window titled "Terminal — tcsh — 80x31". The window has a light blue background and a grey title bar with three colored window control buttons (red, yellow, green) on the left. The terminal text shows a sequence of commands and their outputs. First, a message ";; MSG SIZE rcvd: 213" is displayed. Then, the user runs "nslookup mail.liafa.jussieu.fr". The output shows the DNS server as 194.254.199.85 and the address as 194.254.199.85#53. It then provides a non-authoritative answer for mail.liafa.jussieu.fr, showing its canonical name as liafa1.liafa.jussieu.fr and its IP address as 132.227.93.1. Next, the user runs "nslookup 132.227.93.1". The output shows the same DNS server and address. It then provides a non-authoritative answer for 1.93.227.132.in-addr.arpa, showing its canonical name as liafa1.liafa.jussieu.fr. Finally, it lists authoritative answers that can be found from various nameservers, including 93.227.132.in-addr.arpa, isis.lip6.fr, osiris.lip6.fr, and soleil.uvsq.fr, along with their respective IP and AAAA addresses. The prompt "[poil-aux-oreilles:~] yunes%" is visible at the bottom.

```
;; MSG SIZE rcvd: 213

[poil-aux-oreilles:~] yunes% nslookup mail.liafa.jussieu.fr
Server:          194.254.199.85
Address:         194.254.199.85#53

Non-authoritative answer:
mail.liafa.jussieu.fr canonical name = liafa1.liafa.jussieu.fr.
Name:   liafa1.liafa.jussieu.fr
Address: 132.227.93.1

[poil-aux-oreilles:~] yunes% nslookup 132.227.93.1
Server:          194.254.199.85
Address:         194.254.199.85#53

Non-authoritative answer:
1.93.227.132.in-addr.arpa name = liafa1.liafa.jussieu.fr.

Authoritative answers can be found from:
93.227.132.in-addr.arpa nameserver = liafa1.liafa.jussieu.fr.
93.227.132.in-addr.arpa nameserver = isis.lip6.fr.
93.227.132.in-addr.arpa nameserver = soleil.uvsq.fr.
93.227.132.in-addr.arpa nameserver = osiris.lip6.fr.
isis.lip6.fr internet address = 132.227.60.2
isis.lip6.fr has AAAA address 2001:660:3302:283c::2
liafa1.liafa.jussieu.fr internet address = 132.227.93.1
osiris.lip6.fr internet address = 132.227.60.30
osiris.lip6.fr has AAAA address 2001:660:3302:283c::1e
soleil.uvsq.fr internet address = 193.51.24.1

[poil-aux-oreilles:~] yunes%
```

- D'autre part, sur une machine donnée plusieurs communications peuvent prendre place simultanément
- par conséquent il doit être possible de les distinguer, comme pour différencier les téléphones fixes dans une entreprise, il existe la notion de **port**
 - toute communication nécessite l'utilisation d'un port
 - les ports sont donc utilisés pour assurer le multiplexage, un flux est identifié par deux couples (adresse,port)
- Les ports UDP et TCP sont distincts (deux espaces)

- Il existe essentiellement trois types de ports :
 - les ports reconnus, de numéro compris entre 0 et 1023
 - les ports réservés, de numéro compris entre 1024 et 49151
 - les ports libres, de numéro compris entre 49152 et 65535

- Les **ports reconnus** (Well-known ports) sont utilisés par des services réseau d'usage général et commun :
 - 20 et 21 pour FTP
 - 25 pour SMTP
 - 80 pour HTTP
- ce qui signifie que pour établir une connexion avec un serveur web, il faut s'adresser au port 80 de la machine concernée

- Les **ports réservés** (Registered ports) :
 - certains correspondent à des services d'usage moins général (souvent des services propriétaires)
 - 17500 pour Dropbox
 - n'importe quelle application peut en faire l'usage quelle désire...
- Les **ports libres** (Dynamic, private or ephemeral ports)
 - normalement utilisés pour des durées limitées...



Extrait du fichier `/etc/services`

- sur les machines de la famille Unix, la configuration réseau (d'une interface) peut-être obtenue par la commande `ifconfig`
- `ifconfig` : **configuration** des **interfaces**
- une machine peut posséder plusieurs interfaces

Introduction

boucle locale

tunnel

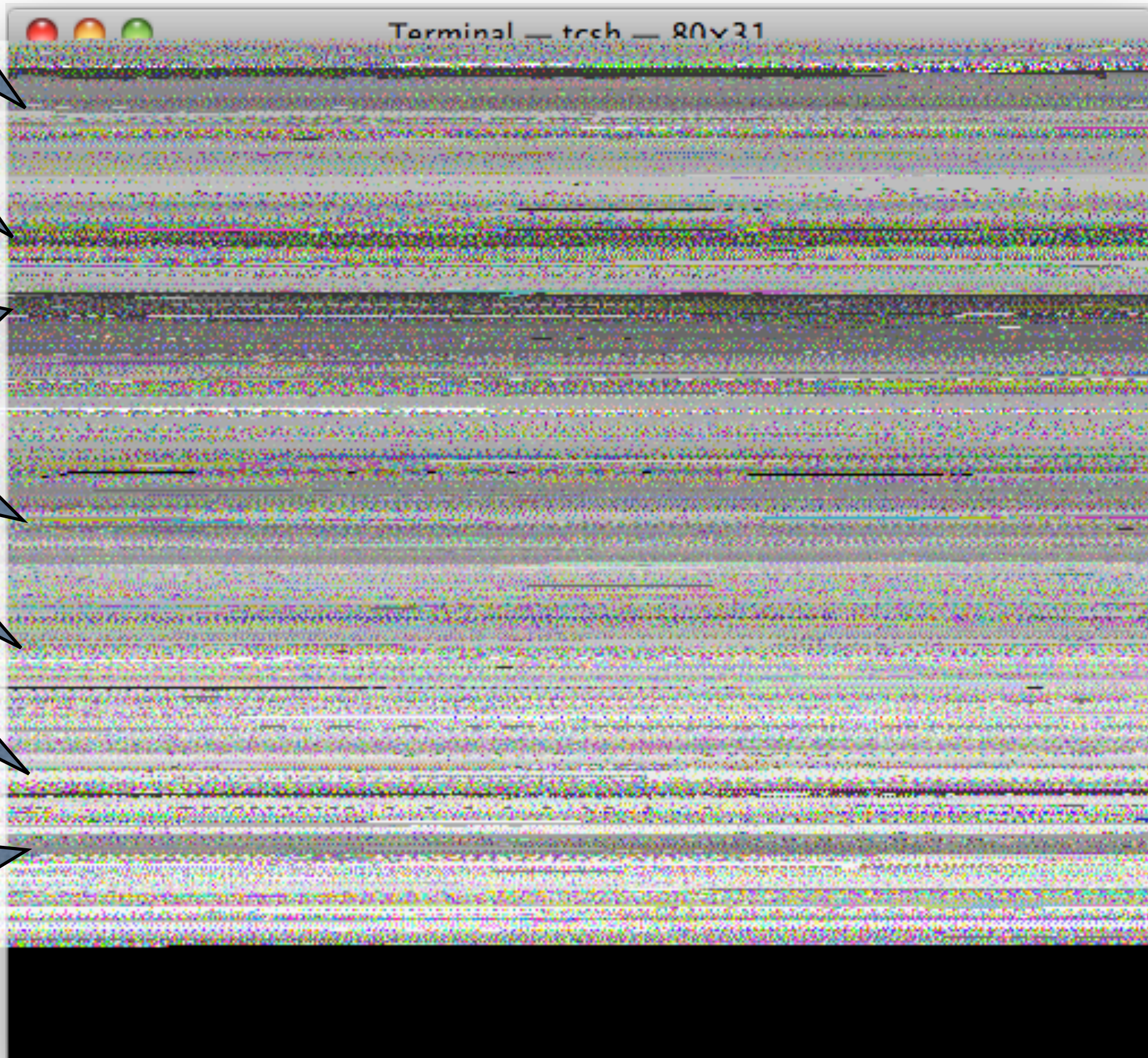
ethernet

WiFi

Firewire

Virtual Box

VMWare



- Pour communiquer entre deux machines distantes séparées par des appareillages de relais, il est nécessaire de trouver un chemin (on dit une **route**)
- il existe donc sur le réseau de quoi permettre le guidage (on dit le **routage**) des paquets
- la consultation (sous Unix) des tables locales de routage s'effectuent à l'aide de la commande netstat



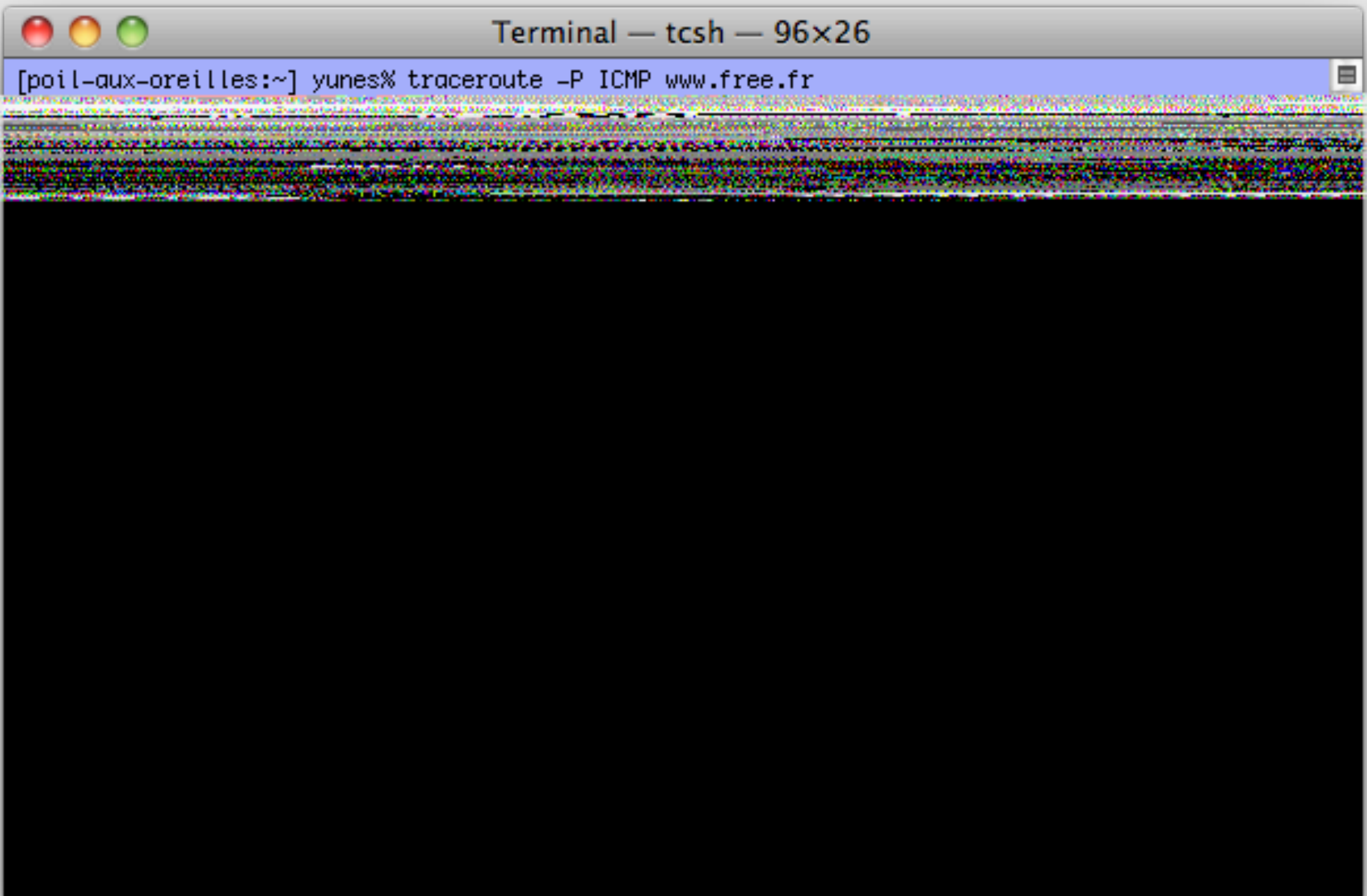
```
Terminal — tcsh — 80x46
[poil-aux-oreilles:~] yunes% netstat -r
Routing tables

Internet:
Destination          Gateway             Flag      Refs      Use   Net46  Exp
```

- On peut observer les routes suivies par des paquets sur le réseau par l'intermédiaire de la commande

tracert [-P protocole] machine
- Attention tracert ne fournit qu'une route probable
- Chaque paquet peut suivre une route différente pour atteindre un point donné à partir d'une même source
- L'algorithme utilisé par tracert ne permet pas d'obtenir une route réellement utilisée (en théorie non, en pratique oui)

Introduction



A terminal window titled "Terminal — tcsh — 96x26" is shown. The prompt is "[poil-aux-oreilles:~] yunes%". The command "traceroute -P ICMP www.free.fr" has been entered. The output of the command is obscured by a large black rectangular redaction box.

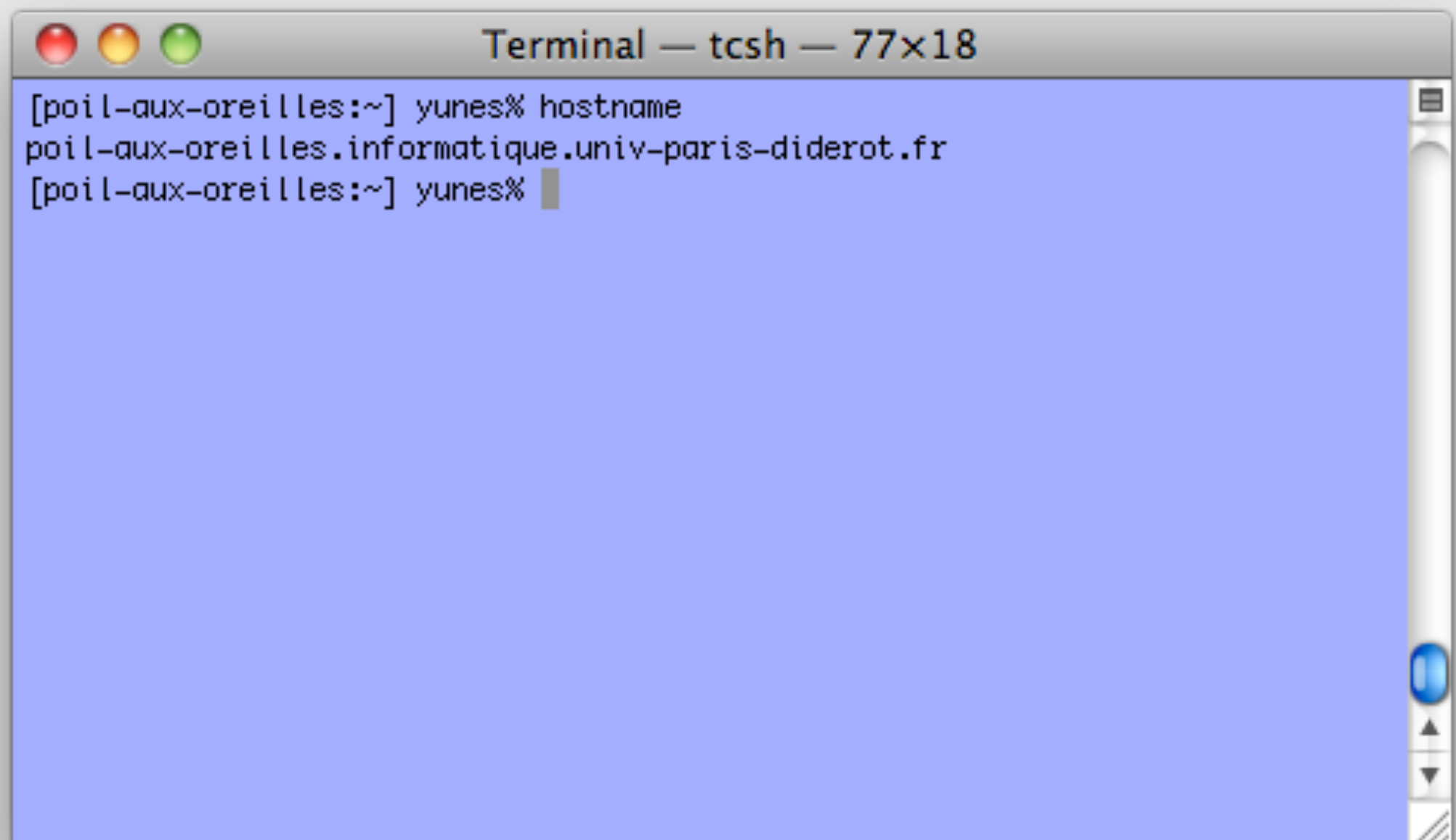
```
Terminal — tcsh — 96x26
[poil-aux-oreilles:~] yunes% traceroute -P ICMP www.free.fr
[Redacted output]
```

- On peut tester si une machine est présente (si le service n'est pas interdit pour des raisons de sécurité) sur le réseau en utilisant le service de très bas-niveau d'écho réseau. Ce service est habituellement désigné sous le vocable ping.
- Ce service peut-être obtenu à l'aide de la commande ping

```
Terminal — tcsh — 83x35

[poil-aux-oreilles:~] yunes% ping www.informatique.univ-paris-diderot.fr
PING trotinette.informatique.univ-paris-diderot.fr (194.254.199.80): 56 data bytes
64 bytes from 194.254.199.80: icmp_seq=0 ttl=255 time=0.340 ms
64 bytes from 194.254.199.80: icmp_seq=1 ttl=255 time=0.472 ms
64 bytes from 194.254.199.80: icmp_seq=2 ttl=255 time=0.355 ms
64 bytes from 194.254.199.80: icmp_seq=3 ttl=255 time=0.411 ms
64 bytes from 194.254.199.80: icmp_seq=4 ttl=255 time=0.390 ms
^C
--- trotinette.informatique.univ-paris-diderot.fr ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.340/0.394/0.472/0.047 ms
[poil-aux-oreilles:~] yunes% ping www.liafa.jussieu.fr
PING liafa1.liafa.jussieu.fr (132.227.93.1): 56 data bytes
64 bytes from 132.227.93.1: icmp_seq=0 ttl=249 time=2.198 ms
64 bytes from 132.227.93.1: icmp_seq=1 ttl=249 time=1.863 ms
64 bytes from 132.227.93.1: icmp_seq=2 ttl=249 time=1.934 ms
64 bytes from 132.227.93.1: icmp_seq=3 ttl=249 time=1.906 ms
64 bytes from 132.227.93.1: icmp_seq=4 ttl=249 time=1.904 ms
^C
--- liafa1.liafa.jussieu.fr ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 1.863/1.961/2.198/0.121 ms
[poil-aux-oreilles:~] yunes% ping www.aliceadsl.fr
PING www.aliceadsl.fr (212.27.48.10): 56 data bytes
64 bytes from 212.27.48.10: icmp_seq=0 ttl=119 time=2.451 ms
64 bytes from 212.27.48.10: icmp_seq=1 ttl=119 time=2.612 ms
64 bytes from 212.27.48.10: icmp_seq=2 ttl=119 time=2.308 ms
64 bytes from 212.27.48.10: icmp_seq=3 ttl=119 time=2.493 ms
64 bytes from 212.27.48.10: icmp_seq=4 ttl=119 time=2.509 ms
64 bytes from 212.27.48.10: icmp_seq=5 ttl=119 time=3.041 ms
^C
--- www.aliceadsl.fr ping statistics ---
6 packets transmitted, 6 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 2.308/2.569/3.041/0.230 ms
[poil-aux-oreilles:~] yunes%
```


- La commande `hostname` permet d'obtenir le nom de la machine sur laquelle on est connecté

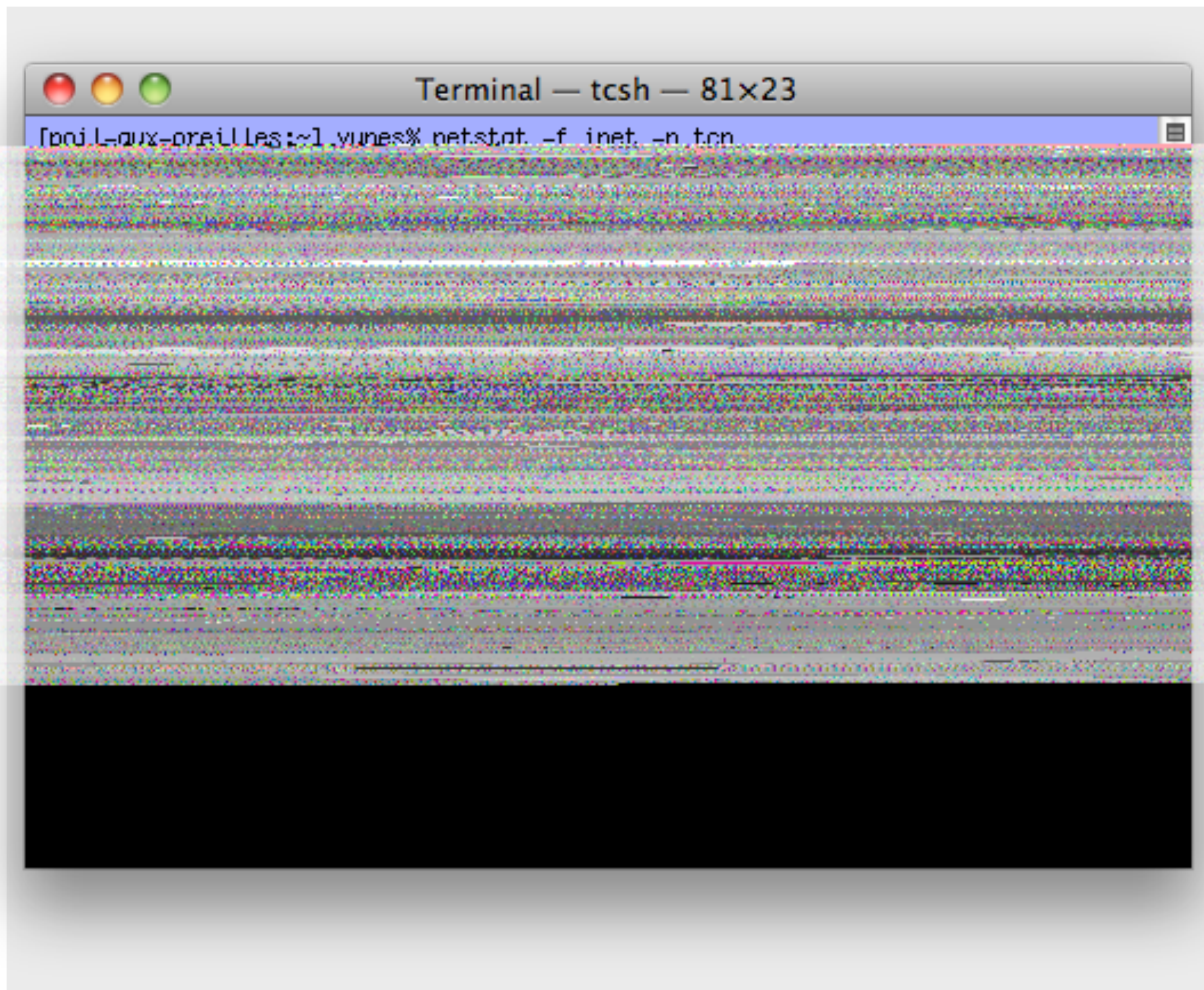


A screenshot of a macOS Terminal window titled "Terminal — tcsh — 77x18". The window has a light blue background. The prompt is "[poil-aux-oreilles:~] yunes%". The command "hostname" has been entered, and the output "poil-aux-oreilles.informatique.univ-paris-diderot.fr" is displayed on the next line. The prompt "[poil-aux-oreilles:~] yunes%" is shown again on the third line, followed by a cursor. The window has standard macOS window controls (red, yellow, green buttons) at the top left and a scroll bar on the right.

```
[poil-aux-oreilles:~] yunes% hostname
poil-aux-oreilles.informatique.univ-paris-diderot.fr
[poil-aux-oreilles:~] yunes%
```

- La commande netstat permet aussi d'obtenir des informations relatives aux connexions connues de la machine

```
netstat [-a] [-f famille] [-p protocole]
```

- Il existe une commande permettant d'établir une liaison TCP « interactive » (i.e. dont l'entrée standard et la sortie standard correspondent à un bout de la liaison)

```
telnet nom_ou_adresse [port]
```

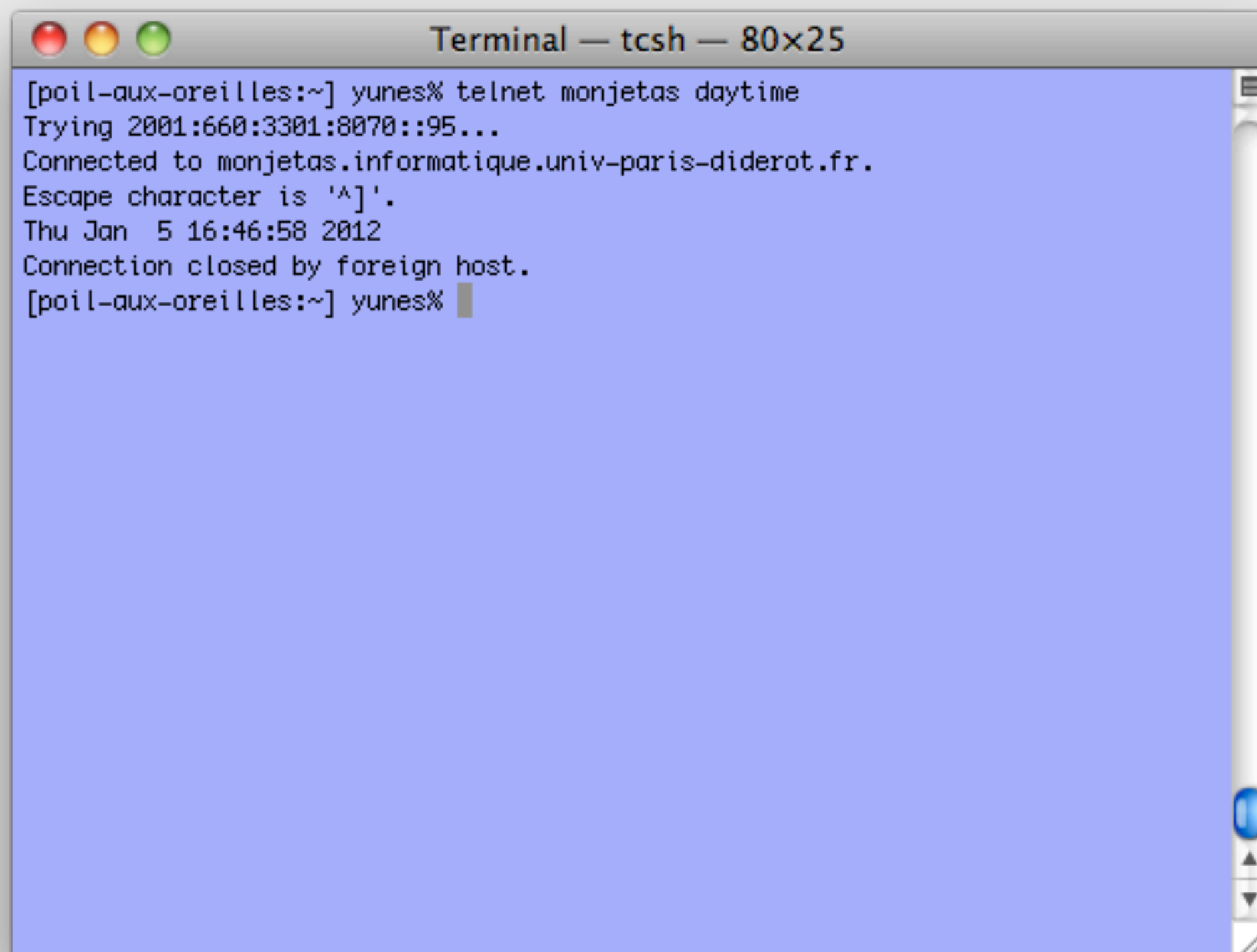

- on établit ici une connexion sur le service echo tcp d'un serveur

clavier

```
Terminal — tcsh — 80x25
[poil-aux-oreilles:~] yunes% telnet monjetas 7
Trying 2001:660:3301:8070::95...
Connected to monjetas.informatique.univ-paris-diderot.fr.
Escape character is '^]'.
BONJOUR LA COMPAGNIE
BONJOUR LA COMPAGNIE
Anacoluthie!
Anacoluthie!
Ectoplasme!
Ectoplasme!
Hangup
[poil-aux-oreilles:~] yunes%
```

réponse
du
serveur

- On établit ici une liaison vers le service daytime en tcp



```
Terminal — tcsh — 80x25
[poil-aux-oreilles:~] yunes% telnet monjetas daytime
Trying 2001:660:3301:8070::95...
Connected to monjetas.informatique.univ-paris-diderot.fr.
Escape character is '^]'.
Thu Jan  5 16:46:58 2012
Connection closed by foreign host.
[poil-aux-oreilles:~] yunes%
```

- L'établissement d'une liaison UDP peut être aussi obtenue par emploi de la commande netcat

```
nc -u machine port
```

- Ces deux commandes permettent de fabriquer des applications réseau à moindre coût
- mais elles seront nécessairement primitives

- Le principe général de l'établissement d'une connexion réseau est
 - que l'un des deux points de communication $[m,p]$ doit être placé en attente d'une demande de connexion (mode serveur)
 - l'autre point de communication $[m',p']$ peut alors demander l'établissement d'une connexion (mode client)
 - la partie serveur décide d'accepter ou non la connexion et si oui
 - un nouveau point de communication $[m,p'']$ est construit (à l'aide d'un port éphémère) et connecté avec le point de connexion du client
 - une connexion est alors établie entre $[m',p']$ et $[m,p'']$
 - à ce moment les deux points sont fonctionnellement interchangeables (symétrie de la liaison)

- L'objet standard de communication est la **socket**
- de quoi s'agit-il ?
 - il faut avoir à l'esprit que dans l'univers inspiré d'Unix « tout est fichier »
 - un abus de langage pour signifier l'uniformité d'opérations d'entrées/sorties : « tout se manipule de la même manière en matière d'entrées/sorties »
 - y compris le réseau... où l'objet sous-jacent correspondant est appelé **socket**



- Une communication standard s'établit de façon suivante :
- la partie serveur est passive et attend toute demande de connexion sur un port donné
- la partie cliente choisit un port depuis lequel elle va émettre sa demande de connexion
- le serveur accepte la demande en établissant la connexion sur un port de service
- lorsque tout est établi, on aura bien un couple identifiant la connexion...

Le codage

- Puisqu'il s'agit de communiquer entre machines diverses et variées, il est nécessaire de rappeler que la représentation des nombres aussi est variée. Pour les entiers, il y a deux grands types de codage :
 - petit-boutiste ou petit-boutien (little-endian)
 - grand-boutiste ou grand-boutien (big-endian)

- On rappelle que le codage (univoque) d'un nombre n en base b s'écrit :

$$n = \sum c_i b^i$$

- comme les machines utilisent des octets, on peut considérer la base comme égale à 256
- pour un entier de 32 bits, 4 octets l'écriture est donc :

$$n = c_3.256^3 + c_2.256^2 + c_1.256^1 + c_0.256^0$$

- se pose donc le problème du stockage de ces 4 octets en mémoire, deux grandes possibilités (parmi d'autres) :

- petit-boutien (petit bout d'abord) :

adresse	m	m+1	m+2	m+3
chiffre	c ₀	c ₁	c ₂	c ₃

- grand-boutien (grand bout d'abord) :

adresse	m	m+1	m+2	m+3
chiffre	c ₃	c ₂	c ₁	c ₀

- dans les communications ce problème est parfois dénommé « the NUXI problem »...
- le protocole IP a donc nécessité de faire un choix :
 - c'est celui du grand-boutisme
 - cet ordre est aussi dénommé NBO (**N**etwork **B**yte **O**rdering)

- il peut être nécessaire d'effectuer soi-même, les conversions du codage courant vers NBO :
- en C, il existe des 4 fonctions pour convertir depuis l'hôte vers le réseau (ou vice-versa) les types entiers de 16 bits et entiers de 32 bits appelés respectivement (short et long) :

```
#include <arpa/inet.h>
```

```
uint32_t htonl(uint32_t hostlong);
uint16_t htons(uint16_t hostshort);
uint32_t ntohl(uint32_t netlong);
uint16_t ntohs(uint16_t netshort);
```


- en C il faut bien prendre soin de réaliser les conversions pour toute donnée utilisée par les protocoles réseau...
- en Java, ce n'est pas nécessaire, l'API fait la conversion elle-même
- car le codage des entiers en Java est standardisé