

Programmation Réseau



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

UFR Informatique

2013-2014

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 donc des ressources pour sa gestion

- 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

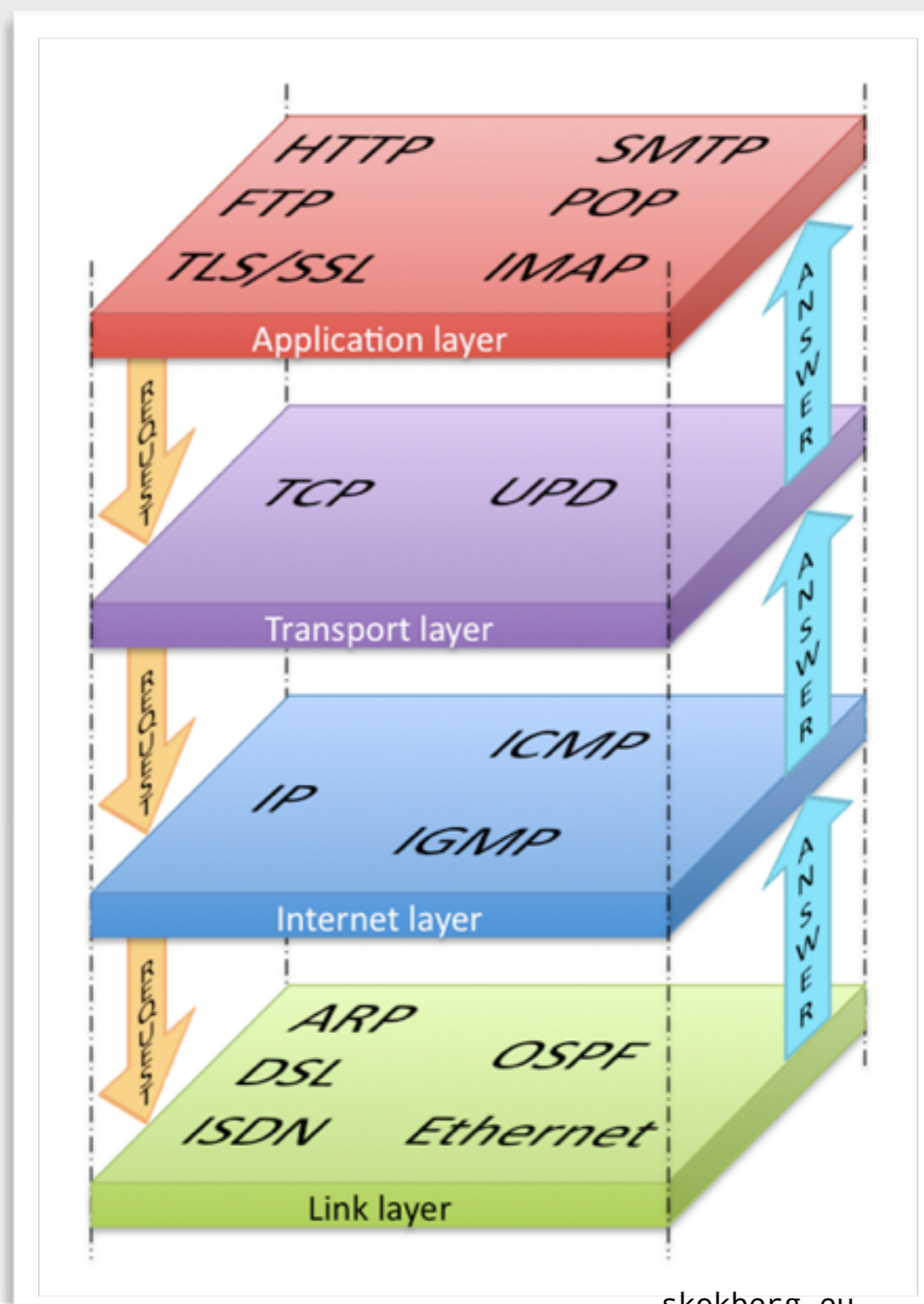
- le modèle de référence est le modèle **ISO/OSI** (**O**pen **S**ystem **I**nterconnection) en 7 couches

OSI (Open Source Interconnection) 7 Layer Model

Layer	Application/Example	Central Device/ Protocols	DOD4 Model
Application (7) Serves as the window for users and application processes to access the network services.	End User layer Program that opens what was sent or creates what is to be sent Resource sharing • Remote file access • Remote printer access • Directory services • Network management	User Applications SMTP	G A T E W A Y Process
Presentation (6) Formats the data to be presented to the Application layer. It can be viewed as the "Translator" for the network.	Syntax layer encrypt & decrypt (if needed) Character code translation • Data conversion • Data compression • Data encryption • Character Set Translation	JPEG/ASCII EBDIC/TIFF/GIF PICT	
Session (5) Allows session establishment between processes running on different stations.	Synch & send to ports (logical ports) Session establishment, maintenance and termination • Session support - perform security, name recognition, logging, etc.	Logical Ports RPC/SQL/NFS NetBIOS names	
Transport (4) Ensures that messages are delivered error-free, in sequence, and with no losses or duplications.	TCP Host to Host, Flow Control Message segmentation • Message acknowledgement • Message traffic control • Session multiplexing	F I L T E R I N G PACKET	Host to Host
Network (3) Controls the operations of the subnet, deciding which physical path the data takes.	Packets ("letter", contains IP address) Routing • Subnet traffic control • Frame fragmentation • Logical-physical address mapping • Subnet usage accounting		Routers IP/IPX/ICMP
Data Link (2) Provides error-free transfer of data frames from one node to another over the Physical layer.	Frames ("envelopes", contains MAC address) [NIC card — Switch — NIC card] (end to end) Establishes & terminates the logical link between nodes • Frame traffic control • Frame sequencing • Frame acknowledgment • Frame delimiting • Frame error checking • Media access control	Switch Bridge WAP PPP/SLIP	Can be used on all layers Network
Physical (1) Concerned with the transmission and reception of the unstructured raw bit stream over the physical medium.	Physical structure Cables, hubs, etc. Data Encoding • Physical medium attachment • Transmission technique - Baseband or Broadband • Physical medium transmission Bits & Volts	Hub Land Based Layers	

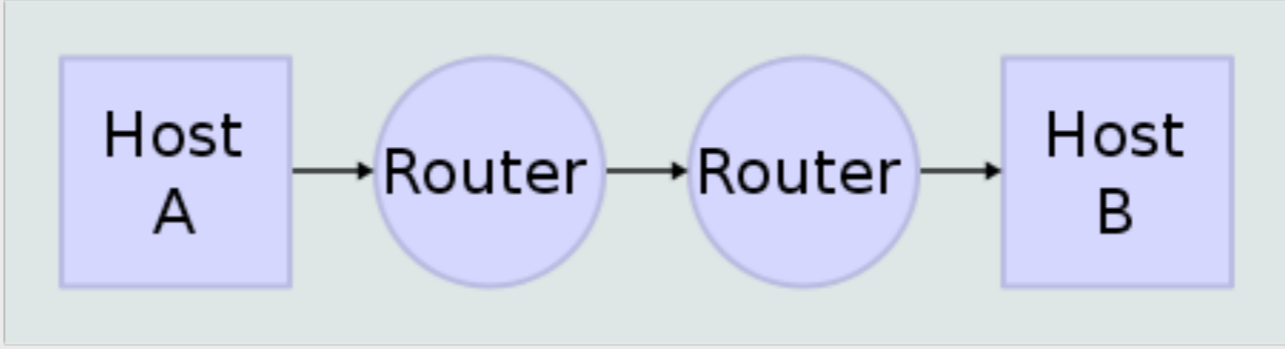
Couche	Protocoles
Application	FTP, HTTP, SMTP, POP, SSH, TELNET, IMAP...
Présentation	SSL, WEP, WPA, Kerberos
Session	Ports
Transport	TCP, UDP, SPX
Réseau	IPv4, IPv6, ARP, IPX
Liaison	802.11, WiFi, ATM, Ethernet, ISDN
Physique	Fibre, Câble, Radio...

- Le modèle Internet est à 4 couches

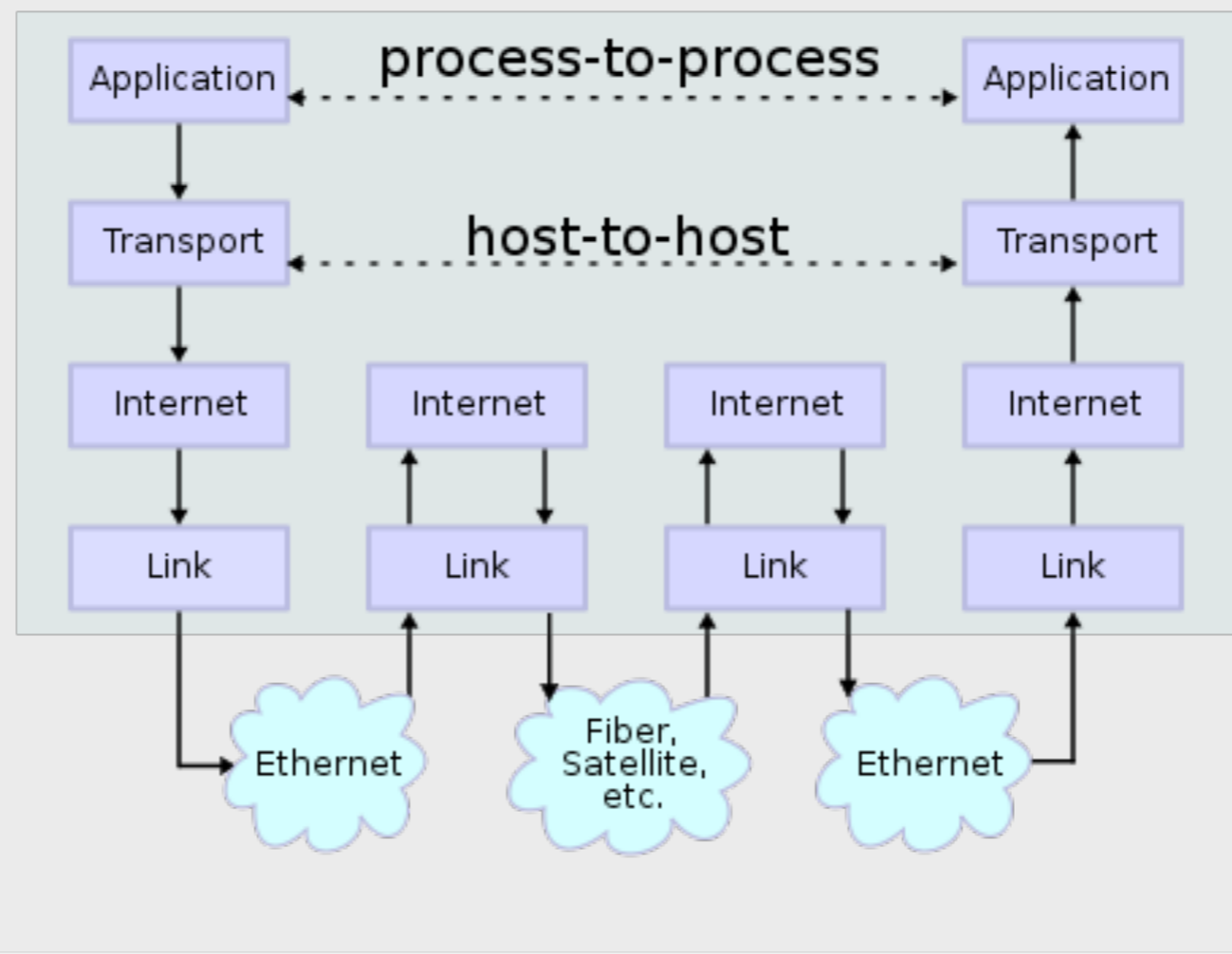


Couche	Protocoles
Application	FTP, HTTP, IMAP, POP...
Transport	TCP, UDP...
Internet	IPv4, IPv6, IPsec, ICMP...
Liaison	ARP, PPP, DSL, Ethernet...

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)
- documents officiels recouvrant tous les aspects de l'Internet
 - <http://www.ietf.org/rfc.html>

- 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...

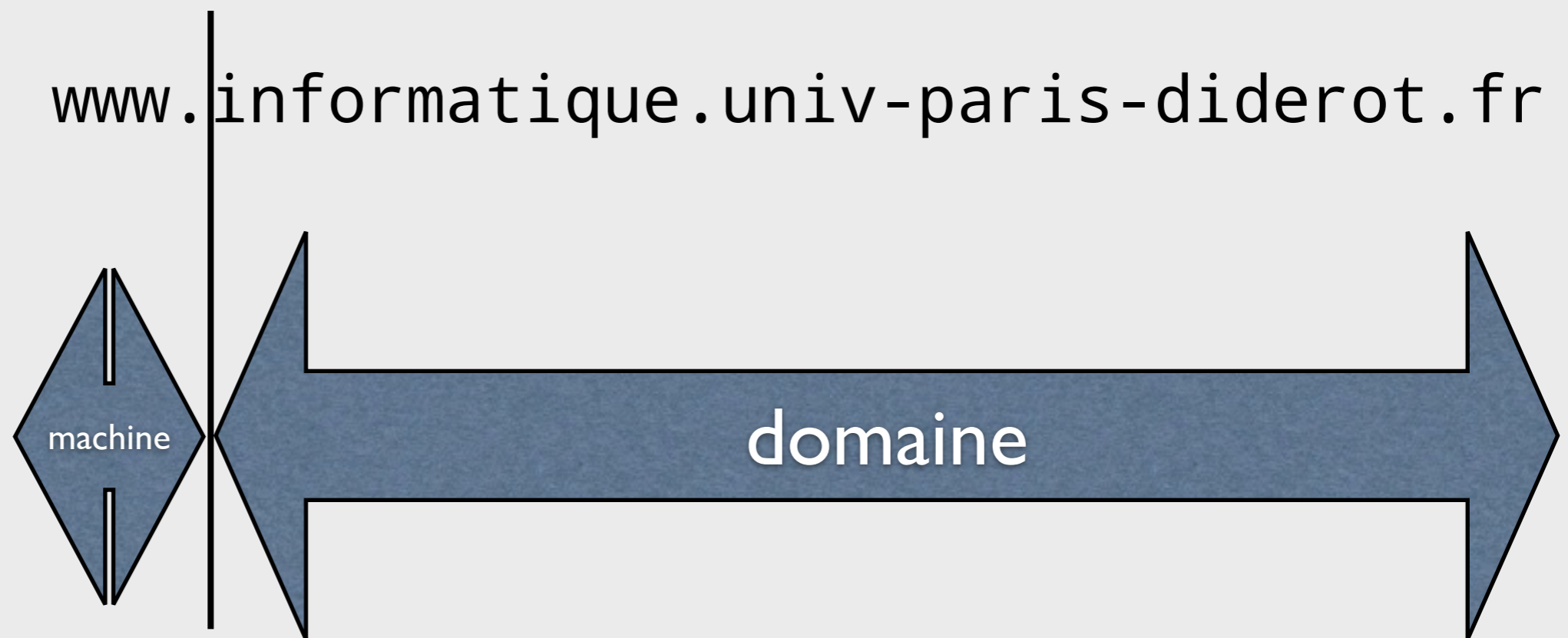
- Le couche transport fournit un ou plusieurs services de communication entre applications (par exemple TCP ou UDP)
 - pour communiquer, ces applications doivent se connaître
 - il existe donc un mécanisme de **nommage**...
- une application fournit un service particulier sur une machine donnée
 - le 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 d'informatique
- `www` est le nom d'une des machines sous la responsabilité de l'UFR d'informatique

- 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...

- 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)

```
Terminal — tcsh — 80x31
[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%
```

```
Terminal — tcsh — 80x31

;; 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...



Terminal — more — 80x31

```
daytime      13/udp      # Daytime (RFC 867)
daytime      13/tcp      # Daytime (RFC 867)
qotd         17/udp      # Quote of the Day
qotd         17/tcp      # Quote of the Day
msp          18/udp      # Message Send Protocol
msp          18/tcp      # Message Send Protocol
chargen     19/udp      # Character Generator
chargen     19/tcp      # Character Generator
ftp-data    20/udp      # File Transfer [Default Data]
ftp-data    20/tcp      # File Transfer [Default Data]
ftp         21/udp      # File Transfer [Control]
ftp         21/tcp      # File Transfer [Control]
ssh         22/udp      # SSH Remote Login Protocol
ssh         22/tcp      # SSH Remote Login Protocol
telnet      23/udp      # Telnet
telnet      23/tcp      # Telnet
            24/udp      # any private mail system
            24/tcp      # any private mail system
smtp        25/udp      # Simple Mail Transfer
smtp        25/tcp      # Simple Mail Transfer
nsw-fe      27/udp      # NSW User System FE
nsw-fe      27/tcp      # NSW User System FE
msg-icp     29/udp      # MSG ICP
msg-icp     29/tcp      # MSG ICP
msg-auth    31/udp      # MSG Authentication
msg-auth    31/tcp      # MSG Authentication
dsp         33/udp      # Display Support Protocol
dsp         33/tcp      # Display Support Protocol
            35/udp      # any private printer server
            35/tcp      # any private printer server
```

⌘

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

```
Terminal — tcsh — 80x31
[poil-aux-oreilles:~] yunes% ifconfig -a
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    inet6 ::1 prefixlen 128
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
    inet 127.0.0.1 netmask 0xff000000
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 00:26:bb:6a:99:d6
    inet6 fe80::226:bbff:fe6a:99d6%en0 prefixlen 64 scopeid 0x4
    inet6 2001:660:3301:8070:226:bbff:fe6a:99d6 prefixlen 64 autoconf
    inet 194.254.199.67 netmask 0xfffff00 broadcast 194.254.199.255
    media: autoselect (1000baseT <full-duplex>)
    status: active
en1: flags=8823<UP,BROADCAST,SMART,SIMPLEX,MULTICAST> mtu 1500
    ether 00:26:bb:15:5f:d2
    media: autoselect (<unknown type>)
    status: inactive
fw0: flags=8822<BROADCAST,SMART,SIMPLEX,MULTICAST> mtu 4078
    lladdr 00:26:bb:ff:fe:6a:99:d6
    media: autoselect <full-duplex>
    status: inactive
vboxnet0: flags=8842<BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 0a:00:27:00:00:00
vmnet1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 00:50:56:c0:00:01
    inet 172.16.42.1 netmask 0xfffff00 broadcast 172.16.42.255
vmnet8: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 00:50:56:c0:00:08
    inet 172.16.191.1 netmask 0xfffff00 broadcast 172.16.191.255
[poil-aux-oreilles:~] yunes%
```

- 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          Flags           Refs      Use  Netif  Expire
default          el-mohino.informat UGSc           10        0    en0
169.254          link#4          UCS             0         0    en0
172.16.42/24     link#8          UC              1         0    vmnet1
172.16.42.255   ff:ff:ff:ff:ff:ff UHLWbI         0         3    vmnet1
172.16.191/24   link#9          UC              1         0    vmnet8
172.16.191.255  link#9          UHLWbI         1         4    vmnet8
194.254.199     link#4          UCS            20         0    en0
mickael.informatiq 0:21:85:3f:8:36  UHLWI          0         64   en0   1105
gaufre.informatiqu d4:85:64:60:62:b2 UHLWI          0         0    en0   1076
ectoplasme.informa 0:14:4f:4a:4:ce  UHLWI          0         0    en0   1157
chicago.informatiq 0:1f:f3:3e:b2:39 UHLWI          0         0    en0   1146
guy-lux.informatiq 60:eb:69:3e:b8:6e UHLWI          0         0    en0   1199
secretariat.inform 0:22:19:d5:81:87 UHLWI          0         0    en0   1166
nivose.informatiqu 0:b:5d:e5:89:42  UHLWI          0         0    en0   1077
pluviose.informati 0:3:ba:27:1a:51  UHLWI          0         0    en0   1168
germinal.informati 0:22:19:d3:c2:72 UHLWI          0         0    en0   1148
messidor.informati 0:1b:24:93:4c:f2 UHLWI          0         0    en0   1127
relents.informatiq 0:3:ba:12:d4:3a  UHLWI          0         0    en0   1184
sweet-smoke.inform 0:21:28:57:42:72 UHLWI          0         12   en0   1185
el-mohino.informat 0:22:83:8b:a4:81 UHLWI          9         6    en0   1197
conejo.informatiqu 8:0:9:c5:4c:2a  UHLWI          0         16   en0   1061
aigreurs.informati 0:25:64:3b:9f:57 UHLWI          0         0    en0   1094
pa-amb-tomaquet.in 0:1b:24:93:4d:7d UHLWI          1         30   en0   1064
194.254.199.255   ff:ff:ff:ff:ff:ff UHLWbI         0         4    en0

Internet6:
Destination      Gateway          Flags           Netif  Expire
default          fe80::222:83ff:fe8 UGSc           en0
localhost        localhost        UH             lo0
dolores-net-v6.ufr link#4          UC             en0
el-mohino.informat 0:22:83:8b:a4:81  UHLW           en0
2001:660:3301:8070 0:26:bb:6a:99:d6  UHL            lo0
fe80::%lo0         localhost        Uc             lo0
localhost        link#1          UHL            lo0
fe80::%en0         link#4          UC             en0
fe80::222:83ff:fe8 0:22:83:8b:a4:81  UHLW           en0
trotinette.local  0:26:bb:6a:99:d6  UHL            lo0
fe80::%en1        link#5          UC             en1
ff01::            localhost        Um             lo0
ff02::            localhost        UmC            lo0
[poil-aux-oreilles:~] yunes%
```

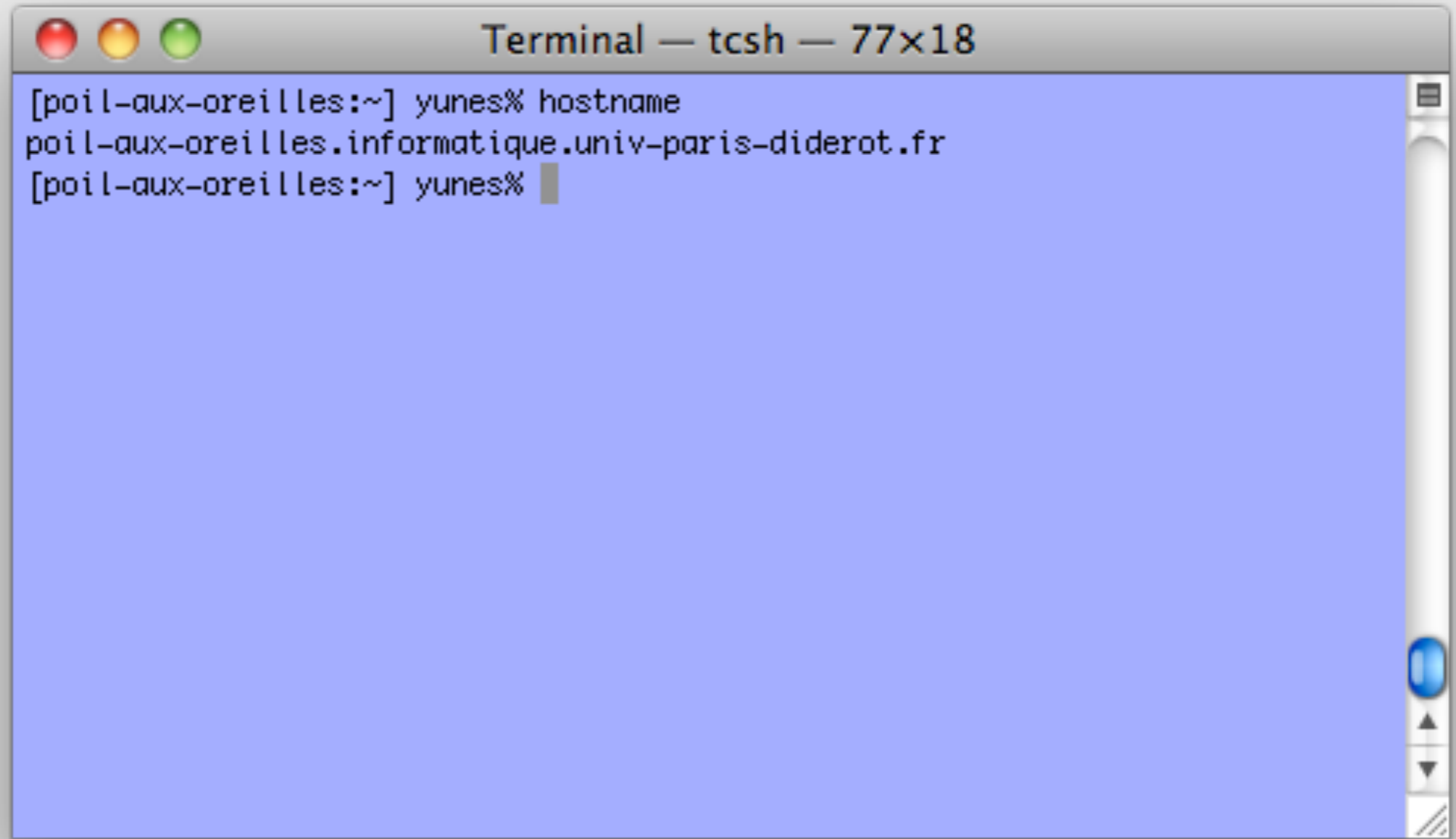
- On peut observer les routes suivies par des paquets sur le réseau par l'intermédiaire de la commande `tracert [-P protocole] machine`
- Attention `tracert` ne fournit qu'une route probable
 - Chaque paquet peut suivre une route différente pour atteindre un point donné à partir d'une même source
 - L'algorithme utilisé par `tracert` ne permet pas d'obtenir une route réellement utilisée (en théorie non, en pratique oui)


```
Terminal — tcsh — 96x26
[poil-aux-oreilles:~] yunes% traceroute -P ICMP www.free.fr
traceroute to www.free.fr (212.27.48.10), 64 hops max, 72 byte packets
 1  el-mohino (194.254.199.88)  0.794 ms  0.558 ms  0.494 ms
 2  backbone-p7.r-prg-1.net.univ-paris7.fr (194.254.200.130)  1.229 ms  1.012 ms  1.042 ms
 3  up7prg.sw-prg-gm.net.univ-paris7.fr (194.254.200.226)  23.671 ms  25.778 ms  3.659 ms
 4  interco-7.01-jussieu.rap.prd.fr (195.221.126.241)  1.195 ms  1.097 ms  1.231 ms
 5  vl165-te3-2-jussieu-rtr-021.noc.renater.fr (193.51.181.102)  1.532 ms  1.319 ms  1.551 ms
 6  te1-2-paris1-rtr-021.noc.renater.fr (193.51.189.230)  2.014 ms  1.789 ms  2.134 ms
 7  aub-6k-1.routers.proxad.net (212.27.38.205)  2.027 ms  * *
 8  th2-crs16-1-be1007.intf.routers.proxad.net (212.27.50.137)  2.503 ms  2.595 ms  2.642 ms
 9  bzn-6k-sys-po21.intf.routers.proxad.net (212.27.50.6)  2.481 ms  2.313 ms  2.643 ms
10  www.free.fr (212.27.48.10)  2.594 ms  2.599 ms  2.528 ms
[poil-aux-oreilles:~] yunes%
```

- 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%
```

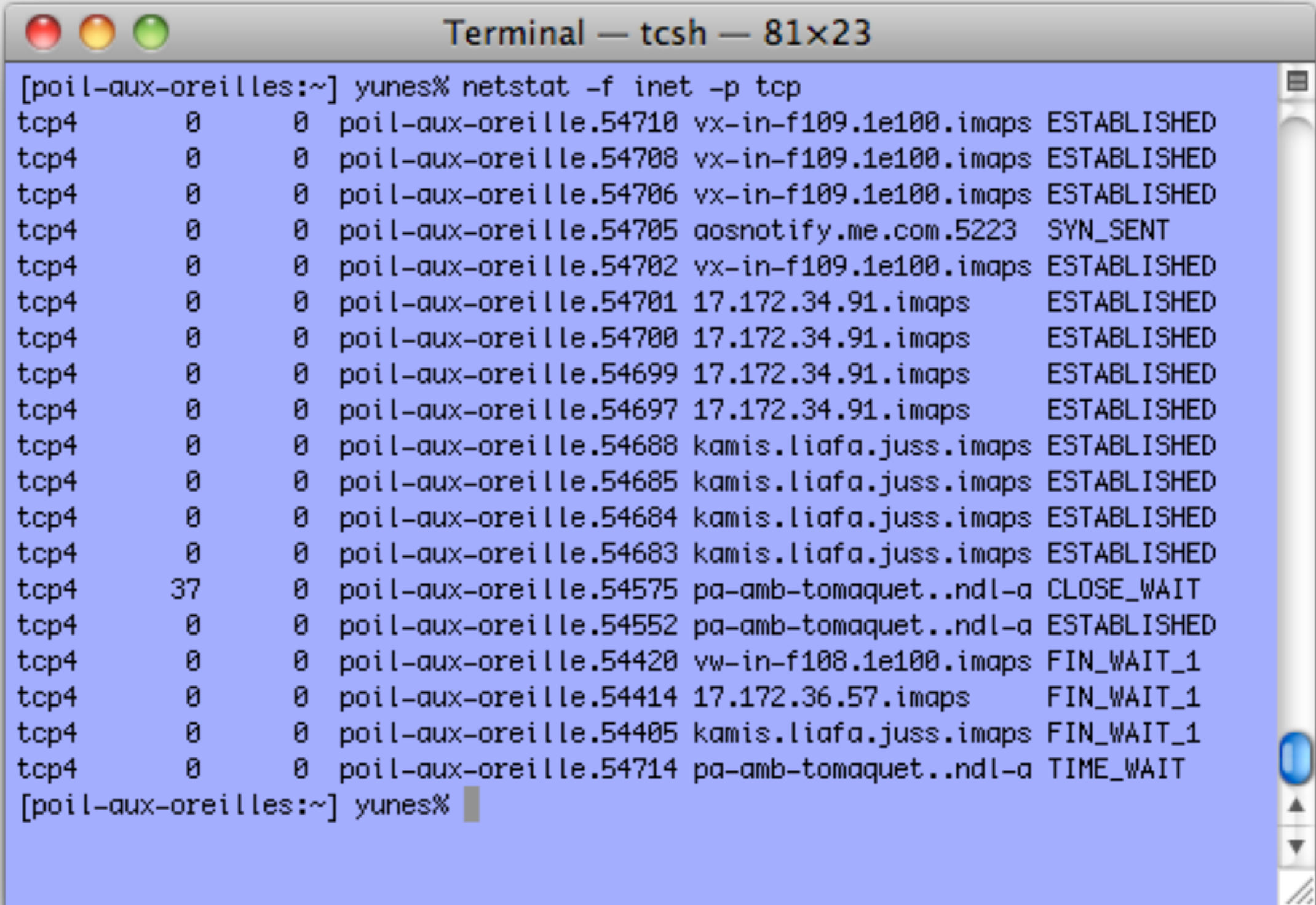
- La commande `hostname` permet d'obtenir le nom de la machine sur laquelle on est connecté



```
Terminal — tcsh — 77x18
[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]
```



```
Terminal — tcsh — 81x23
[poil-aux-oreilles:~] yunes% netstat -f inet -p tcp
tcp4      0      0  poil-aux-oreille.54710 vx-in-f109.1e100.imaps ESTABLISHED
tcp4      0      0  poil-aux-oreille.54708 vx-in-f109.1e100.imaps ESTABLISHED
tcp4      0      0  poil-aux-oreille.54706 vx-in-f109.1e100.imaps ESTABLISHED
tcp4      0      0  poil-aux-oreille.54705 aosnotify.me.com.5223  SYN_SENT
tcp4      0      0  poil-aux-oreille.54702 vx-in-f109.1e100.imaps ESTABLISHED
tcp4      0      0  poil-aux-oreille.54701 17.172.34.91.imaps    ESTABLISHED
tcp4      0      0  poil-aux-oreille.54700 17.172.34.91.imaps    ESTABLISHED
tcp4      0      0  poil-aux-oreille.54699 17.172.34.91.imaps    ESTABLISHED
tcp4      0      0  poil-aux-oreille.54697 17.172.34.91.imaps    ESTABLISHED
tcp4      0      0  poil-aux-oreille.54688 Kamis.liafa.juss.imaps ESTABLISHED
tcp4      0      0  poil-aux-oreille.54685 Kamis.liafa.juss.imaps ESTABLISHED
tcp4      0      0  poil-aux-oreille.54684 Kamis.liafa.juss.imaps ESTABLISHED
tcp4      0      0  poil-aux-oreille.54683 Kamis.liafa.juss.imaps ESTABLISHED
tcp4      37     0  poil-aux-oreille.54575 pa-amb-tomaquet..ndl-a CLOSE_WAIT
tcp4      0      0  poil-aux-oreille.54552 pa-amb-tomaquet..ndl-a ESTABLISHED
tcp4      0      0  poil-aux-oreille.54420 vw-in-f108.1e100.imaps FIN_WAIT_1
tcp4      0      0  poil-aux-oreille.54414 17.172.36.57.imaps    FIN_WAIT_1
tcp4      0      0  poil-aux-oreille.54405 Kamis.liafa.juss.imaps FIN_WAIT_1
tcp4      0      0  poil-aux-oreille.54714 pa-amb-tomaquet..ndl-a TIME_WAIT
[poil-aux-oreilles:~] yunes%
```


- 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]
```

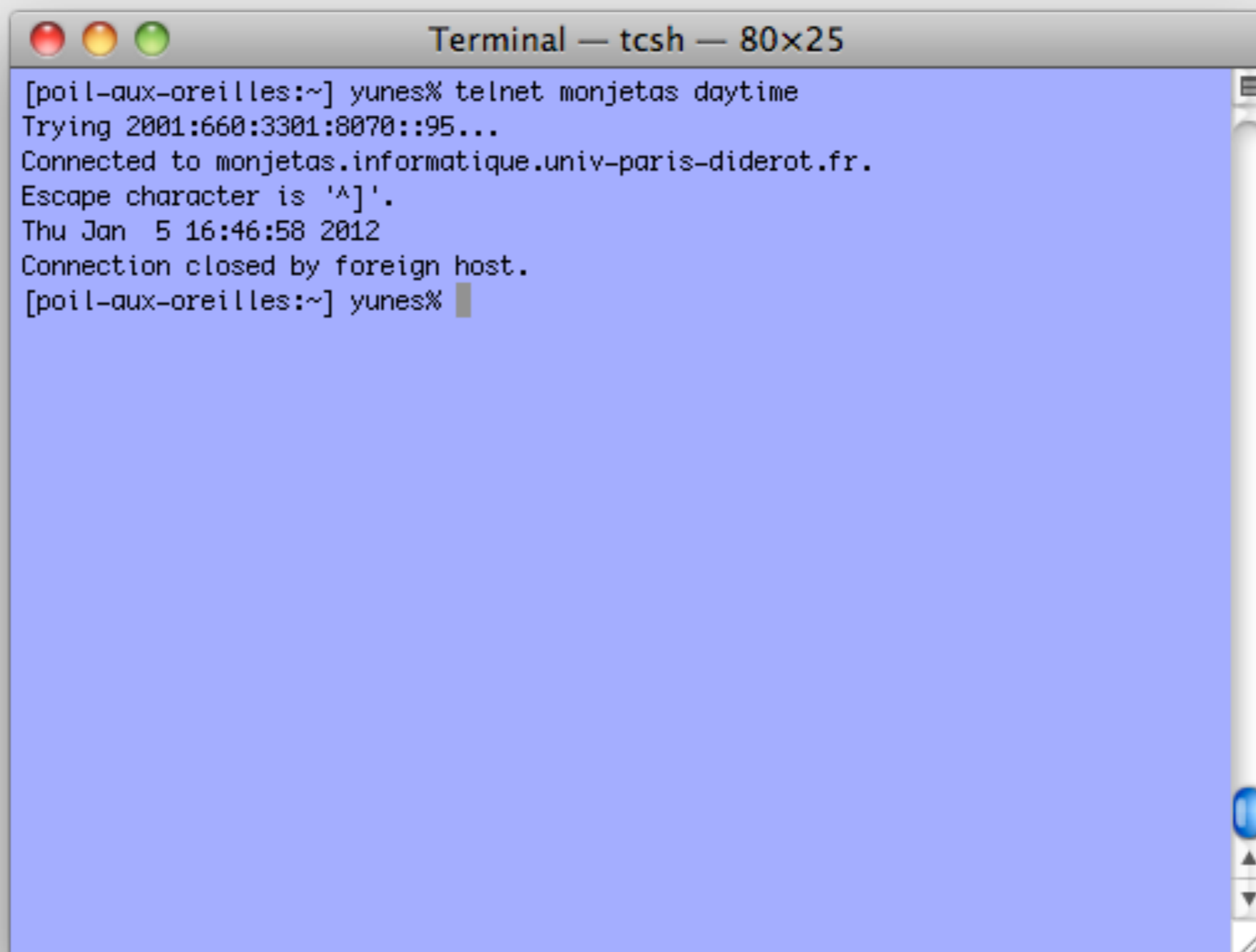

- on établit ici une connexion sur le service echo tcp d'un serveur

clavier

réponse
du
serveur

```
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
Anacoluthé!
Anacoluthé!
Ectoplasme!
Ectoplasme!
Hangup
[poil-aux-oreilles:~] yunes%
```

- 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%
```

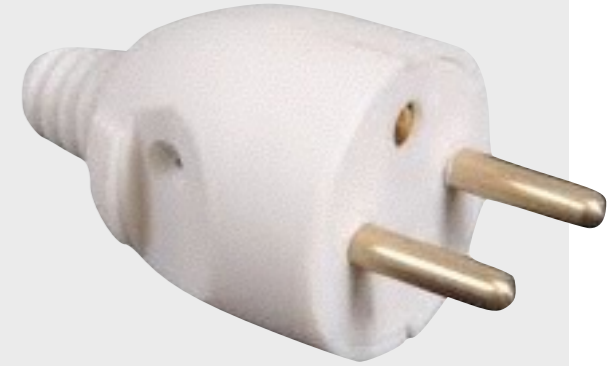
- L'établissement d'une liaison UDP peut être aussi obtenue par emploi de la commande netcat

```
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	c	c	c

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

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

- 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