RAPPORT N°1

TP 1

1. INTERFACES RESEAU ET ADRESSE IP

OBSERVEZ LES INTERFACES ETH0 ET LO DE VOTRE MACHINE AVEC /SBIN/IFCONFIG. INDIQUEZ LES ADRESSES IPV4 ET IPV6 CORRESPONDANTES, LA TAILLE DE LA PARTIE RESEAU, LES TAILLES MAXIMALES DE PAQUETS (MTU, MAXIMUM TRANSMISSION UNIT), ETC. UTILISEZ EGALEMENT LA COMMANDE IP ADDR LS , C'EST LA MEME CHOSE, MAIS EN VERSION PLUS MODERNE.

Avec ifconfig :

```
thmoreau@alesia:~$ /sbin/ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9000
        inet 10.0.103.5  netmask 255.255.255.0  broadcast 10.0.103.255
        inet6 2001:660:6101:800:103::5  prefixlen 80  scopeid 0x0<global>
        inet6 fe80::da9e:f3ff:fe10:2c66  prefixlen 64  scopeid 0x20<link>
        ether d8:9e:f3:10:2c:66  txqueuelen 1000  (Ethernet)
        RX packets 31447  bytes 35596227 (33.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 19156  bytes 2857819 (2.7 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 16  memory 0xef200000-ef220000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Boucle locale)
        RX packets 336  bytes 36443 (35.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 336  bytes 36443 (35.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

On peut voir sur l'interface eth0 l'adresse IPv4 (10.0.103.5), l'adresse IPv6 (fe80:da9e:f3ff:fe10:2c66), la MTU (9000), etc. Pour l'interface lo on a également l'adresse IPv4 (127.0.0.1), l'adresse IPv6 (: : 1), la MTU (65536), etc.

Avec ip addr ls :

```
thmoreau@alesia:~$ ip addr ls
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc pfifo_fast state UP group default qlen 1000
    link/ether d8:9e:f3:10:2c:66 brd ff:ff:ff:ff:ff:ff
    altname enp0s31f6
    inet 10.0.103.5/24 brd 10.0.103.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 2001:660:6101:800:103::5/80 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::da9e:f3ff:fe10:2c66/64 scope link
       valid_lft forever preferred_lft forever
```

POUR LES IPV4, UTILISEZ IPCALC L_ADRESSE_IP/LA_TAILLE_DU_PREFIXE POUR OBTENIR LE DETAIL DES ADRESSES APPARTENANT AU MEME RESEAU QUE CELLE DE VOTRE MACHINE.

```
thmoreau@alesia:~$ ipcalc 10.0.103.5/24
Address:   10.0.103.5           00001010.00000000.01100111. 00000101
Netmask:   255.255.255.0 = 24   11111111.11111111.11111111. 00000000
Wildcard:  0.0.0.255            00000000.00000000.00000000. 11111111
=>
Network:   10.0.103.0/24        00001010.00000000.01100111. 00000000
HostMin:   10.0.103.1           00001010.00000000.01100111. 00000001
HostMax:   10.0.103.254         00001010.00000000.01100111. 11111110
Broadcast: 10.0.103.255         00001010.00000000.01100111. 11111111
Hosts/Net: 254                   Class A, Private Internet
```

```
thmoreau@alesia:~$ ipcalc 127.0.0.1/8
Address:   127.0.0.1            01111111. 00000000.00000000.00000001
Netmask:   255.0.0.0 = 8        11111111. 00000000.00000000.00000000
Wildcard:  0.255.255.255        00000000. 11111111.11111111.11111111
=>
Network:   127.0.0.0/8          01111111. 00000000.00000000.00000000
HostMin:   127.0.0.1            01111111. 00000000.00000000.00000001
HostMax:   127.255.255.254      01111111. 11111111.11111111.11111110
Broadcast: 127.255.255.255      01111111. 11111111.11111111.11111111
Hosts/Net: 16777214              Class A, Loopback
```

D'APRES SES ADRESSES IPV4 ET IPV6, A QUOI CORRESPOND L'INTERFACE LO ?

L'interface lo est l'adresse locale, autrement appelé loopback.

OBSERVEZ QUE LA MTU N'EST PAS LA MEME POUR LO ET POUR ETH0, POURQUOI ?

Je n'ai pas trouvé d'informations concrètes à ce sujet.

LA COMMANDE PING UNEIP PERMET DE TESTER LA CONNECTIVITE IP PAR L'EMISSION D'UNE REQUETE ECHO ICMP. VERIFIEZ QUE VOUS ETES BIEN RELIE A LA MACHINE DE VOTRE VOISIN (UTILISER CONTROL-C POUR L'ARRETER). TESTEZ A LA FOIS AVEC UNE IPV4 ET AVEC UNE IPV6.

```
thmoreau@alesia:~$ ping 10.0.103.6
PING 10.0.103.6 (10.0.103.6) 56(84) bytes of data.
64 bytes from 10.0.103.6: icmp_seq=1 ttl=64 time=0.181 ms
64 bytes from 10.0.103.6: icmp_seq=2 ttl=64 time=0.127 ms
64 bytes from 10.0.103.6: icmp_seq=3 ttl=64 time=0.122 ms
64 bytes from 10.0.103.6: icmp_seq=4 ttl=64 time=0.122 ms
64 bytes from 10.0.103.6: icmp_seq=5 ttl=64 time=0.124 ms
64 bytes from 10.0.103.6: icmp_seq=6 ttl=64 time=0.129 ms
64 bytes from 10.0.103.6: icmp_seq=7 ttl=64 time=0.121 ms
64 bytes from 10.0.103.6: icmp_seq=8 ttl=64 time=0.123 ms
64 bytes from 10.0.103.6: icmp_seq=9 ttl=64 time=0.126 ms
64 bytes from 10.0.103.6: icmp_seq=10 ttl=64 time=0.120 ms
```

L'adresse 10.0.103.6 correspond à l'adresse IPv4 de la machine « aluminium ».

```
thmoreau@alesia:~$ ping fe80::da9e:f3ff:fe0d:c9ea
PING fe80::da9e:f3ff:fe0d:c9ea(fe80::da9e:f3ff:fe0d:c9ea) 56 data bytes
64 bytes from fe80::da9e:f3ff:fe0d:c9ea%eth0: icmp_seq=1 ttl=25 time=0.208 ms
64 bytes from fe80::da9e:f3ff:fe0d:c9ea%eth0: icmp_seq=2 ttl=25 time=0.124 ms
64 bytes from fe80::da9e:f3ff:fe0d:c9ea%eth0: icmp_seq=3 ttl=25 time=0.130 ms
64 bytes from fe80::da9e:f3ff:fe0d:c9ea%eth0: icmp_seq=4 ttl=25 time=0.125 ms
64 bytes from fe80::da9e:f3ff:fe0d:c9ea%eth0: icmp_seq=5 ttl=25 time=0.130 ms
64 bytes from fe80::da9e:f3ff:fe0d:c9ea%eth0: icmp_seq=6 ttl=25 time=0.132 ms
64 bytes from fe80::da9e:f3ff:fe0d:c9ea%eth0: icmp_seq=7 ttl=25 time=0.127 ms
64 bytes from fe80::da9e:f3ff:fe0d:c9ea%eth0: icmp_seq=8 ttl=25 time=0.128 ms
64 bytes from fe80::da9e:f3ff:fe0d:c9ea%eth0: icmp_seq=9 ttl=25 time=0.125 ms
64 bytes from fe80::da9e:f3ff:fe0d:c9ea%eth0: icmp_seq=10 ttl=25 time=0.120 ms
```

Celle-ci correspond à son adresse IPv6.

2. NETCAT & NETSTAT

LANCEZ NC -L -P 12345 ET LAISSEZ-LE TOURNER. OBSERVEZ (EN PASSANT PAR UN AUTRE TERMINAL) DANS NETSTAT -TUAN OU SS -TUAN L'APPARITION DU SERVICE PARMI LES AUTRES. NOTE : UTILISEZ GREP POUR RETROUVER PLUS FACILEMENT LE PORT 12345.

Sur un autre terminal, après avoir exécuté la commande netstat -tuan, on peut voir ceci apparaître (une partie seulement car tout ne passait pas sur la capture).

```
thmoreau@alesia:~$ netstat -tuan
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale          Adresse distante        Etat
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:62354           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:38229           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:12345           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:5666            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:873             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:5355            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:59565           0.0.0.0:*               LISTEN
tcp        0      0 10.0.103.5:931          10.0.220.51:2049        ESTABLISHED
tcp6       0      0 :::111                  :::*                    LISTEN
tcp6       0      0 :::4369                 :::*                    LISTEN
tcp6       0      0 :::39539                :::*                    LISTEN
tcp6       0      0 :::59893                :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 ::1:631                 :::*                    LISTEN
tcp6       0      0 :::5666                 :::*                    LISTEN
tcp6       0      0 :::873                  :::*                    LISTEN
tcp6       0      0 :::5355                 :::*                    LISTEN
tcp6       0      0 2001:660:6101:800:10:22 2001:660:6101:800:43778 ESTABLISHED
tcp6       0      0 2001:660:6101:800:10:22 2001:660:6101:800:38064 ESTABLISHED
tcp6       0      0 2001:660:6101:800:10:22 2001:660:6101:800:33166 ESTABLISHED
udp        0      0 127.0.0.1:1004          0.0.0.0:*
udp        0      0 0.0.0.0:33826           0.0.0.0:*
udp        0      0 0.0.0.0:5355            0.0.0.0:*
```

Sur la 7e ligne on peut y voir 0.0.0.0:12345.

AJOUTEZ A NETSTAT OU SS L'OPTION -P POUR CONSTATER QUE C'EST BIEN LE PROGRAMME NC QUI EST A L'ECOUTE. C'EST DONC UN MINI-SERVEUR QUE L'ON A LANCE, AUQUEL ON VA MAINTENANT SE CONNECTER.

```
thmoreau@alesia:~$ netstat -tuan -p
(Tous les processus ne peuvent être identifiés, les infos sur les processus
non possédés ne seront pas affichées, vous devez être root pour les voir toutes.)
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale          Adresse distante        Etat        PID/Program name
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:62354           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:38229           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:12345           0.0.0.0:*               LISTEN      3426849/nc
tcp        0      0 0.0.0.0:5666            0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:873             0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:5355            0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:59565           0.0.0.0:*               LISTEN      -
tcp        0      0 10.0.103.5:931          10.0.220.51:2049        ESTABLISHED -
tcp6       0      0 :::111                  :::*                    LISTEN      -
tcp6       0      0 :::4369                 :::*                    LISTEN      -
tcp6       0      0 :::39539                :::*                    LISTEN      -
tcp6       0      0 :::59893                :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
tcp6       0      0 ::1:631                 :::*                    LISTEN      -
tcp6       0      0 :::5666                 :::*                    LISTEN      -
tcp6       0      0 :::873                  :::*                    LISTEN      -
tcp6       0      0 :::5355                 :::*                    LISTEN      -
tcp6       0      0 2001:660:6101:800:10:22 2001:660:6101:800:43778 ESTABLISHED -
tcp6       0      0 2001:660:6101:800:10:22 2001:660:6101:800:38064 ESTABLISHED -
tcp6       0      0 2001:660:6101:800:10:22 2001:660:6101:800:33166 ESTABLISHED -
```

Le programme nc est donc bien à l'écoute comme le montre cette capture.

DITES A VOTRE VOISIN DE LANCER NC VOTREMACHINE 12345 , POUR JOUER LE ROLE DU CLIENT. OBSERVEZ DANS NETSTAT -TUN OU SS -TU LA CONNEXION ETABLIE ENTRE CLIENT ET SERVEUR.

```
thmoreau@aluminium:~$ nc 10.0.103.5 12345
```

```
thmoreau@alesia:~$ netstat -tun
Connexions Internet actives (sans serveurs)
Proto Recv-Q Send-Q Adresse locale          Adresse distante        Etat
tcp        0      0 10.0.103.5:931          10.0.220.51:2049        ESTABLISHED
tcp        0      0 10.0.103.5:12345        10.0.103.6:54066        ESTABLISHED
tcp6       0      0 2001:660:6101:800:10:22 2001:660:6101:800:43778 ESTABLISHED
tcp6       0      0 2001:660:6101:800:10:22 2001:660:6101:800:38064 ESTABLISHED
tcp6       0      0 2001:660:6101:800:10:22 2001:660:6101:800:33166 ESTABLISHED
```

La connexion a bien été établie entre le client et le serveur.

TAPEZ DES LIGNES D'UN COTE OU DE L'AUTRE, OBSERVEZ QUE C'EST EFFECTIVEMENT TRANSMIS DE L'AUTRE COTE.

```
thmoreau@alesia:~$ nc -l -p 12345
oui
dd
Ici Alesia Bonjour
Salut Alesia moi c'est Aluminium
```

```
thmoreau@aluminium:~$ nc alesia 12345
oui
dd
Ici Alesia Bonjour
Salut Alesia moi c'est Aluminium
```

3. PROTOCOLE ARP

IDENTIFIEZ LES ADRESSES DES MACHINES VOISINES AVEC LESQUELLES DES ECHANGES RECENTS ONT EU LIEU (TABLE ARP, DISPONIBLE PAR LA COMMANDE /USR/SBIN/ARP , ON PEUT UTILISER L'OPTION -N POUR AVOIR LES ADRESSES IP PLUTOT QUE LES NOMS DE MACHINES). IL DOIT Y AVOIR AU MOINS L'ADRESSE DU ROUTEUR (EN .254, ON VERRA DANS LA SECTION ROUTAGE CI-DESSOUS).

```
thmoreau@alesia:~$ /usr/sbin/arp
Adresse                  TypeMap AdresseMat          Indicateurs          Iface
vlan103-routeur.emi.u-b  ether   58:20:b1:b1:23:00   C                    eth0
aluminium.emi.u-bordeau  ether   d8:9e:f3:0d:c9:ea   C                    eth0
```

```
thmoreau@alesia:~$ /usr/sbin/arp -n
Adresse                  TypeMap AdresseMat          Indicateurs          Iface
10.0.103.254             ether   58:20:b1:b1:23:00   C                    eth0
10.0.103.6               ether   d8:9e:f3:0d:c9:ea   C                    eth0
```

VERIFIEZ QUE LORSQUE VOUS EMETTEZ AVEC PING -4 UNE REQUETE ICMP ECHO VERS UNE MACHINE DE LA SALLE QUI NE FIGURE PAS ENCORE DANS VOTRE TABLE ARP, CETTE MACHINE APPARAIT DANS LA TABLE ARP DE VOTRE MACHINE ET LA VOTRE, DANS LA TABLE ARP DE L'AUTRE.

```
thmoreau@alesia:~$ ping -4 nenjetepus
PING nenjetepus.emi.u-bordeaux.fr (10.0.103.19) 56(84) bytes of data.
From alesia.emi.u-bordeaux.fr (10.0.103.5) icmp_seq=1 Destination Host Unreachable
From alesia.emi.u-bordeaux.fr (10.0.103.5) icmp_seq=5 Destination Host Unreachable
From alesia.emi.u-bordeaux.fr (10.0.103.5) icmp_seq=6 Destination Host Unreachable
From alesia.emi.u-bordeaux.fr (10.0.103.5) icmp_seq=8 Destination Host Unreachable
From alesia.emi.u-bordeaux.fr (10.0.103.5) icmp_seq=9 Destination Host Unreachable
```

```
thmoreau@alesia:~$ /usr/sbin/arp
Adresse                  TypeMap AdresseMat          Indicateurs           Iface
vlan103-routeur.emi.u-b  ether   58:20:b1:b1:23:00   C                     eth0
nenjetepus.emi.u-bordea          (incomplete)                              eth0
aluminium.emi.u-bordeau  ether   d8:9e:f3:0d:c9:ea   C                     eth0
```

La machine Nenjetepus n'est pas allumée mais elle figure quand même dans la table ARP après la commande ping.

ESSAYEZ LA COMMANDE IP NEIGH LS ; C'EST LA MEME CHOSE EN VERSION PLUS MODERNE, ET CONTIENT NOTAMMENT AUSSI LES VOISINS EN IPV6.

```
thmoreau@alesia:~$ ip neigh ls
10.0.103.254 dev eth0 lladdr 58:20:b1:b1:23:00 REACHABLE
10.0.103.19 dev eth0  FAILED
10.0.103.6 dev eth0 lladdr d8:9e:f3:0d:c9:ea REACHABLE
2001:660:6101:800:103::6 dev eth0 lladdr d8:9e:f3:0d:c9:ea STALE
fe80::da9e:f3ff:fe0d:c9ea dev eth0 lladdr d8:9e:f3:0d:c9:ea STALE
fe80::5a20:b1ff:feb1:2300 dev eth0 lladdr 58:20:b1:b1:23:00 router REACHABLE
2001:660:6101:800:103::ffff dev eth0 lladdr 58:20:b1:b1:23:00 router REACHABLE
```

ESSAYEZ DE FAIRE UN PING VERS 8.8.8.8 ET VERS 10.0.252.4. POURQUOI LEURS ADRESSES N'APPARAISSENT PAS DANS LA TABLE ARP ?

```
thmoreau@alesia:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=113 time=10.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=113 time=10.1 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=113 time=10.1 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=113 time=9.99 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=113 time=10.0 ms
```

```
thmoreau@alesia:~$ ping 10.0.252.4
PING 10.0.252.4 (10.0.252.4) 56(84) bytes of data.
From 10.0.253.247 icmp_seq=1 Time to live exceeded
From 10.0.253.247 icmp_seq=2 Time to live exceeded
From 10.0.253.247 icmp_seq=3 Time to live exceeded
From 10.0.253.247 icmp_seq=4 Time to live exceeded
From 10.0.253.247 icmp_seq=5 Time to live exceeded
```

D'après mes recherches, l'adresse 8.8.8.8 serait le serveur DNS principal de Google.

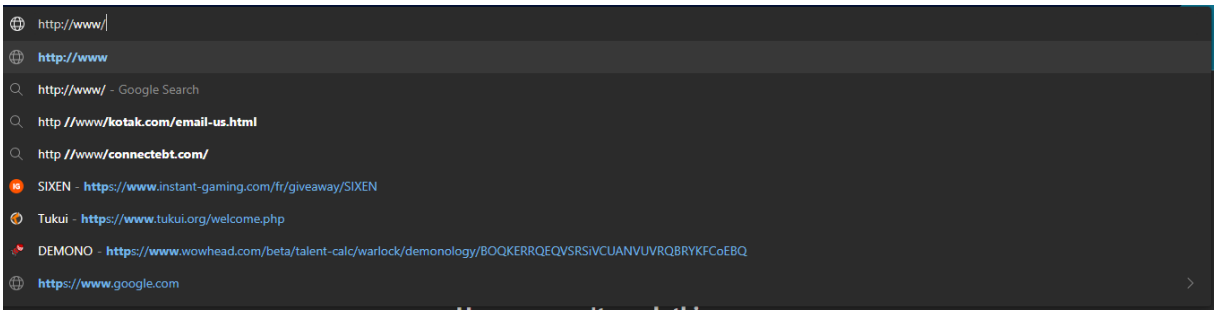
4. RESOLUTION DE NOMS (DNS)

LISEZ LE FICHIER /ETC/RESOLV.CONF ET MAN RESOLV.CONF, POURQUOI Y A-T-IL PLUSIEURS ADRESSES IP ?
LA LIGNE SEARCH PERMET D'EVITER D'AVOIR A TAPER LE NOM DE MACHINE EN ENTIER.
ESSAYEZ DE TAPER HTTP://WWW/ TOUT COURT DANS UN NAVIGATEUR WEB ET OBSERVEZ COMMENT CELA EST COMPLETE POUR CONFIRMER.

```
thmoreau@alesia:~$ cat /etc/resolv.conf
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "resolvectl status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs should typically not access this file directly, but only
# through the symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a
# different way, replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 127.0.0.53
options edns0 trust-ad
search emi.u-bordeaux.fr cremi.emi.u-bordeaux1.fr
```

```
RESOLV.CONF(5)                          Linux Programmer's Manual                          RESOLV.CONF(5)

NAME
       resolv.conf - resolver configuration file

SYNOPSIS
       /etc/resolv.conf

DESCRIPTION
       The  resolver  is  a  set  of routines in the C library that provide access to the Internet Domain Name System
       (DNS).  The resolver configuration file contains information that is read by the resolver routines  the  first
       time they are invoked by a process.  The file is designed to be human readable and contains a list of keywords
       with values that provide various types of resolver  information.   The  configuration  file  is  considered  a
       trusted  source  of  DNS  information  (e.g.,  DNSSEC AD-bit information will be returned unmodified from this
       source).

       If this file does not exist, only the name server on the local machine will be queried, and  the  search  list
       contains the local domain name determined from the hostname.

       The different configuration options are:

       nameserver Name server IP address
              Internet  address of a name server that the resolver should query, either an IPv4 address (in dot nota-
              tion), or an IPv6 address in colon (and possibly dot) notation as per RFC 2373.  Up to MAXNS (currently
              3, see <resolv.h>) name servers may be listed, one per keyword.  If there are multiple servers, the re-
              solver library queries them in the order listed.  If no nameserver entries are present, the default  is
              to  use  the name server on the local machine.  (The algorithm used is to try a name server, and if the
 Manual page resolv.conf(5) line 1 (press h for help or q to quit)
```

Il n’y a qu’une adresse visible ici.



POUR EFFECTUER EXPLICITEMENT UNE RESOLUTION DE NOM, UTILISEZ LA COMMANDE HOST (OU EVENTUELLEMENT NSLOOKUP)

```
thmoreau@alesia:~$ host yahoo.com
yahoo.com has address 98.137.11.163
yahoo.com has address 98.137.11.164
yahoo.com has address 74.6.143.25
yahoo.com has address 74.6.143.26
yahoo.com has address 74.6.231.20
yahoo.com has address 74.6.231.21
yahoo.com has IPv6 address 2001:4998:124:1507::f001
yahoo.com has IPv6 address 2001:4998:24:120d::1:0
yahoo.com has IPv6 address 2001:4998:24:120d::1:1
yahoo.com has IPv6 address 2001:4998:44:3507::8000
yahoo.com has IPv6 address 2001:4998:44:3507::8001
yahoo.com has IPv6 address 2001:4998:124:1507::f000
yahoo.com mail is handled by 1 mta5.am0.yahoodns.net.
yahoo.com mail is handled by 1 mta6.am0.yahoodns.net.
yahoo.com mail is handled by 1 mta7.am0.yahoodns.net.
```

ESSAYEZ DE RESOUDRE YAHOO.COM . POURQUOI Y A-T-IL PLUSIEURS ADRESSES IP ? OBSERVEZ EGALEMENT QUE CELA RETOURNE A LA FOIS DES ADRESSES IPV4 ET DES ADRESSES IPV6. REESSAYEZ PLUSIEURS FOIS. IL PEUT ARRIVER QUE LE RESULTAT SOIT DIFFERENT, POURQUOI ?

Il y a plusieurs machines différentes pour le serveur de Yahoo et elles ne répondront pas forcément toutes, voilà pourquoi le résultat peut différer.

PARFOIS IL EST AUSSI UTILE D'AJOUTER DES NOMS DE MACHINE A LA MAIN, JETEZ UN OEIL AU FICHIER /ETC/HOSTS (ET AU MANUEL). REMARQUEZ DANS /ETC/NSSWITCH.CONF LA LIGNE HOSTS: QUI INDIQUE QUE C'EST LE FICHIER /ETC/HOSTS (FILES) QUI A LA PRIORITE SUR LA RESOLUTION DNS (DNS).

```
thmoreau@alesia:~$ cat /etc/hosts
#HOSTS_VERSION=CREMI_00005
#
#Default :
127.0.0.1       localhost
#127.0.1.1      alesia.emi.u-bordeaux1.fr alesia

# The following lines are desirable for IPv6 capable hosts
::1             ip6-localhost ip6-loopback
fe00::0         ip6-localnet
ff00::0         ip6-mcastprefix
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters

#Following lines are auto-generated by script CREMI/ CFengine3
10.0.103.5 alesia.emi.u-bordeaux.fr alesia alesia
2001:660:6101:800:103::5 alesia.emi.u-bordeaux.fr alesia alesia
10.0.230.25 wheezy.emi.u-bordeaux.fr stretch wheezy
2001:660:6101:800:230::25 wheezy.emi.u-bordeaux.fr stretch wheezy
10.0.230.18 boursouf.emi.u-bordeaux.fr boursouf boursouf
2001:660:6101:800:230::18 boursouf.emi.u-bordeaux.fr boursouf boursouf
10.0.230.19 boursouflet.emi.u-bordeaux.fr boursouflet boursouflet
2001:660:6101:800:230::19 boursouflet.emi.u-bordeaux.fr boursouflet boursouflet
10.0.230.23 cocatris.emi.u-bordeaux.fr cocatris cocatris
2001:660:6101:800:230::23 cocatris.emi.u-bordeaux.fr cocatris cocatris
10.0.230.4 infini1.emi.u-bordeaux.fr infini1 infini1
2001:660:6101:800:230::4 infini1.emi.u-bordeaux.fr infini1 infini1
10.0.230.5 infini2.emi.u-bordeaux.fr infini2 infini2
```

```
thmoreau@alesia:~$ cat /etc/nsswitch.conf
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc-reference' and `info' packages installed, try:
# `info libc "Name Service Switch"' for information about this file.

passwd:         files db
group:          files db
shadow:         files

hosts:          files  dns
networks:       files

protocols:      files
services:       files
ethers:         files
rpc:            files
```

5. SERVICES AU CREMI : LDAP & NFS

POUR L'IDENTIFICATION ET L'AUTHENTIFICATION, C'EST LE PROTOCOLE LDAP QUI EST UTILISE, LA CONFIGURATION EST LISIBLE DANS LE CHIER /ETC/LDAP.CONF (PAS LA PEINE DE TOUT LIRE, CE QUI NOUS INTERESSE EST AU TOUT DEBUT), POURQUOI Y A-T-IL PLUSIEURS SERVEURS (LIGNE HOST) ? OBSERVEZ LES ADRESSES IP DE CES SERVEURS.

```
thmoreau@alesia:~$ cat /etc/ldap.conf
#host cresus dionysos bromios
host ldap-ad dionysos bromios  cresus

base DC=cremi,DC=emi,DC=u-bordeaux1,DC=fr
ldap_version 3
```

TROUVEZ LE NUMERO DE PORT DE CE SERVICE, VERIFIEZ DANS /ETC/SERVICES, RETROUVEZ-Y EGALEMENT LES PORTS HTTP, SSH, X11, ETC. (UTILISEZ GREP !)

```
thmoreau@alesia:~$ grep ldap /etc/services
ldap            389/tcp                 # Lightweight Directory Access Protocol
ldap            389/udp
ldaps           636/tcp                         # LDAP over SSL
ldaps           636/udp
```

```
thmoreau@alesia:~$ grep http /etc/services
# Updated from https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml .
http            80/tcp          www             # WorldWideWeb HTTP
https           443/tcp                         # http protocol over TLS/SSL
http-alt        8080/tcp        webcache        # WWW caching service
```

```
thmoreau@alesia:~$ grep ssh /etc/services
ssh             22/tcp                          # SSH Remote Login Protocol
```

```
thmoreau@alesia:~$ grep x11 /etc/services
x11             6000/tcp        x11-0           # X Window System
x11-1           6001/tcp
x11-2           6002/tcp
x11-3           6003/tcp
x11-4           6004/tcp
x11-5           6005/tcp
x11-6           6006/tcp
x11-7           6007/tcp
```

VOS FICHIERS SONT STOCKES SUR UN SERVEUR NFS, UTILISEZ LA COMMANDE DF ~ POUR REPERER LE NOM DU SERVEUR, LE CHEMIN SUR LE SERVEUR, ET LE CHEMIN OU CELA APPARAIT SUR VOTRE MACHINE

```
thmoreau@alesia:~$ df ~
Sys. de fichiers      blocs de 1K Utilisé Disponible Uti% Monté sur
unityaccount:/account     9437184 1522176    7915008  17% /autofs/unityaccount/cremi
```

TP 2

1. CONFIGURATION D'UN RESEAU LOCAL

A L'AIDE DE LA COMMANDE IFCONFIG -A (MAN IFCONFIG), DONNEZ LA LISTE DES INTERFACES RESEAUX

```
eth0: flags=4098<BROADCAST,MULTICAST>  mtu 1500
        ether aa:aa:aa:aa:00:00  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1  (Local Loopback)
        RX packets 40  bytes 2720 (2.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 40  bytes 2720 (2.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Les interfaces réseau sont eth0 et lo.

ON DECIDE DE CONFIGURER L'INTERFACE ETH0 DE TELLE SORTE QUE LA MACHINE IMMORTAL POSSEDE L'ADRESSE 192.168.0.1/24. QUELLE EST L'ADRESSE DU RESEAU ? QUEL EST LE MASQUE DU RESEAU ? QUELLE EST LA PLAGE D'ADRESSE IP DE CE RESEAU.

L'adresse du réseau 192.168.0.0.
Le masque du réseau est 255.255.255.0.
La plage d'adresse IP est de 192.168.0.1 (en ne prenant pas en compte l'adresse réseau) à 192.168.0.254 (broadcast est exclu).

CONFIGUREZ IMMORTAL A L'AIDE DE LA COMMANDE IFCONFIG ETH0 192.168.0.1/24 UP. CONFIGUREZ DE MANIERE ANALOGUE LES 3 AUTRES MACHINES.

L'adresse 192.168.0.1 est associée à immortal
192.168.0.2 pour opeth
192.168.0.3 pour syl
192.168.0.4 pour grave

```
root@immortal:~# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.1  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a8aa:aaff:feaa:0  prefixlen 64  scopeid 0x20<link>
        ether aa:aa:aa:aa:00:00  txqueuelen 1000  (Ethernet)
        RX packets 66  bytes 6240 (6.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 74  bytes 6888 (6.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1  (Local Loopback)
        RX packets 220  bytes 17840 (17.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 220  bytes 17840 (17.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
root@opeth:~# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.2  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a8aa:aaff:feaa:100  prefixlen 64  scopeid 0x20<link>
        ether aa:aa:aa:aa:01:00  txqueuelen 1000  (Ethernet)
        RX packets 24  bytes 2200 (2.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 30  bytes 2728 (2.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1  (Local Loopback)
        RX packets 40  bytes 2624 (2.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 40  bytes 2624 (2.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
root@syl:~# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.3  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a8aa:aaff:feaa:200  prefixlen 64  scopeid 0x20<link>
        ether aa:aa:aa:aa:02:00  txqueuelen 1000  (Ethernet)
        RX packets 24  bytes 2200 (2.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 30  bytes 2728 (2.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1  (Local Loopback)
        RX packets 40  bytes 2560 (2.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 40  bytes 2560 (2.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
root@grave:~# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.4  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a8aa:aaff:feaa:300  prefixlen 64  scopeid 0x20<link>
        ether aa:aa:aa:aa:03:00  txqueuelen 1000  (Ethernet)
        RX packets 24  bytes 2200 (2.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 30  bytes 2728 (2.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1  (Local Loopback)
        RX packets 40  bytes 2624 (2.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 40  bytes 2624 (2.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

VERIFIEZ VOS CONFIGURATIONS A L'AIDE DE LA COMMANDE PING. QUEL EST LE PROTOCOLE UTILISE PAR LE PROGRAMME PING ?

```
root@immortal:~# ping 192.168.0.2
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.
64 bytes from 192.168.0.2: icmp_seq=1 ttl=64 time=0.191 ms
64 bytes from 192.168.0.2: icmp_seq=2 ttl=64 time=0.626 ms
64 bytes from 192.168.0.2: icmp_seq=3 ttl=64 time=0.612 ms
64 bytes from 192.168.0.2: icmp_seq=4 ttl=64 time=0.505 ms
64 bytes from 192.168.0.2: icmp_seq=5 ttl=64 time=0.629 ms
^C
--- 192.168.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 0.191/0.512/0.629/0.167 ms
```

La commande ping utilise le protocole ICMP.

METTEZ EN EVIDENCE QUE LE PING FONCTIONNE A L'AIDE DE LA COMMANDE TCPDUMP -I ETH0 QUI PERMET D'AFFICHER TOUT LE TRAFIC RESEAU ENTRANT ET SORTANT D'UNE CERTAINE MACHINE (SUR L'INTERFACE ETH0).

```
root@opeth:~# tcpdump -i eth0
[ 1877.932588] device eth0 entered promiscuous mode
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
23:08:55.485418 IP 192.168.0.1 > 192.168.0.2: ICMP echo request, id 588, seq 1, length 64
23:08:55.485434 IP 192.168.0.2 > 192.168.0.1: ICMP echo reply, id 588, seq 1, length 64
23:08:56.484666 IP 192.168.0.1 > 192.168.0.2: ICMP echo request, id 588, seq 2, length 64
23:08:56.484699 IP 192.168.0.2 > 192.168.0.1: ICMP echo reply, id 588, seq 2, length 64
23:08:57.486699 IP 192.168.0.1 > 192.168.0.2: ICMP echo request, id 588, seq 3, length 64
23:08:57.486736 IP 192.168.0.2 > 192.168.0.1: ICMP echo reply, id 588, seq 3, length 64
23:08:58.488666 IP 192.168.0.1 > 192.168.0.2: ICMP echo request, id 588, seq 4, length 64
23:08:58.488699 IP 192.168.0.2 > 192.168.0.1: ICMP echo reply, id 588, seq 4, length 64
23:08:59.490703 IP 192.168.0.1 > 192.168.0.2: ICMP echo request, id 588, seq 5, length 64
23:08:59.490740 IP 192.168.0.2 > 192.168.0.1: ICMP echo reply, id 588, seq 5, length 64
```

On peut voir que Opeth (192.168.0.2) reçoit bien les requêtes d'Immortal (192.168.0.1).

ESSAYEZ DE PINGUER UNE IP AUTRE QUE CELLES CONFIGUREES. ON VOIT PASSER DES REQUETES ARP, ON EN REPARLERA PLUS TARD.

(Ne fonctionne pas)

ESSAYEZ UN PING AVEC L'ADRESSE DE BROADCAST DU RESEAU. QUE SE PASSE-T-IL ? LES REQUETES SONT-ELLES REÇUES PAR TOUTES LES MACHINES ? EST-CE QU'ELLES Y REPONDENT ? CORRIGEZ LE PROBLEME EN TAPANT CETTE COMMANDE SUR TOUS LES POSTES :
SYSCTL NET.IPV4.ICMP_ECHO_IGNORE_BROADCASTS=0
RE-ESSAYER UN PING AVEC L'ADRESSE DE BROADCAST DU RESEAU. QUE CONSTATEZ-VOUS ?

```
root@immortal:~# ping 192.168.0.255
ping: Do you want to ping broadcast? Then -b. If not, check your local firewall rules
root@immortal:~# -b
-bash: -b: command not found
root@immortal:~# ping 192.168.0.255 -b
WARNING: pinging broadcast address
PING 192.168.0.255 (192.168.0.255) 56(84) bytes of data.
```

A première vue la commande semble fonctionner, malgré une alerte.

```
root@opeth:~# tcpdump -i eth0
[ 2706.017398] device eth0 entered promiscuous mode
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
23:22:38.779520 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 674, seq 37, length 64
23:22:38.958577 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 674, seq 38, length 64
23:22:39.966608 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 674, seq 39, length 64
23:22:40.974609 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 674, seq 40, length 64
23:22:41.982559 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 674, seq 41, length 64
```

```
root@syl:~# tcpdump -i eth0
[ 2744.712548] device eth0 entered promiscuous mode
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
23:23:17.442375 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 674, seq 76, length 64
23:23:18.211489 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 674, seq 77, length 64
23:23:19.219406 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 674, seq 78, length 64
23:23:20.227521 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 674, seq 79, length 64
```

```
root@grave:~# tcpdump -i eth0
[ 2762.391171] device eth0 entered promiscuous mode
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
23:23:35.119566 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 674, seq 93, length 64
23:23:35.319922 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 674, seq 94, length 64
23:23:36.327757 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 674, seq 95, length 64
23:23:37.335898 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 674, seq 96, length 64
```

Et comme on peut le voir chaque machine reçoit les requêtes, cependant elles n'y répondent pas.

```
PING 192.168.0.255 (192.168.0.255) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=0.018 ms
64 bytes from 192.168.0.2: icmp_seq=1 ttl=64 time=0.833 ms (DUP!)
64 bytes from 192.168.0.4: icmp_seq=1 ttl=64 time=1.84 ms (DUP!)
64 bytes from 192.168.0.3: icmp_seq=1 ttl=64 time=1.96 ms (DUP!)
64 bytes from 192.168.0.1: icmp_seq=2 ttl=64 time=0.040 ms
64 bytes from 192.168.0.4: icmp_seq=2 ttl=64 time=3.66 ms (DUP!)
64 bytes from 192.168.0.2: icmp_seq=2 ttl=64 time=3.68 ms (DUP!)
64 bytes from 192.168.0.3: icmp_seq=2 ttl=64 time=3.69 ms (DUP!)
64 bytes from 192.168.0.1: icmp_seq=3 ttl=64 time=0.040 ms
64 bytes from 192.168.0.2: icmp_seq=3 ttl=64 time=3.89 ms (DUP!)
64 bytes from 192.168.0.3: icmp_seq=3 ttl=64 time=3.90 ms (DUP!)
64 bytes from 192.168.0.4: icmp_seq=3 ttl=64 time=3.91 ms (DUP!)
64 bytes from 192.168.0.1: icmp_seq=4 ttl=64 time=0.037 ms
64 bytes from 192.168.0.4: icmp_seq=4 ttl=64 time=3.19 ms (DUP!)
64 bytes from 192.168.0.2: icmp_seq=4 ttl=64 time=3.21 ms (DUP!)
64 bytes from 192.168.0.3: icmp_seq=4 ttl=64 time=3.21 ms (DUP!)
```

Après avoir exécuté la commande sysctl net.ipv4.icmp_echo_ignore_broadcasts=0 sur chaque machine on peut voir ceci sur la machine émettrice.

```
root@opeth:~# tcpdump -i eth0
[ 3217.531473] device eth0 entered promiscuous mode
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
23:31:10.290290 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 678, seq 106, length 64
23:31:10.290324 IP 192.168.0.2 > 192.168.0.1: ICMP echo reply, id 678, seq 106, length 64
23:31:10.492980 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 678, seq 107, length 64
23:31:10.493004 IP 192.168.0.2 > 192.168.0.1: ICMP echo reply, id 678, seq 107, length 64
23:31:11.494422 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 678, seq 108, length 64
23:31:11.494458 IP 192.168.0.2 > 192.168.0.1: ICMP echo reply, id 678, seq 108, length 64
23:31:12.495708 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 678, seq 109, length 64
23:31:12.495747 IP 192.168.0.2 > 192.168.0.1: ICMP echo reply, id 678, seq 109, length 64
```

```
root@syl:~# tcpdump -i eth0
[ 3234.979214] device eth0 entered promiscuous mode
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
23:31:27.710449 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 678, seq 124, length 64
23:31:27.710482 IP 192.168.0.3 > 192.168.0.1: ICMP echo reply, id 678, seq 124, length 64
23:31:28.463639 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 678, seq 125, length 64
23:31:28.463673 IP 192.168.0.3 > 192.168.0.1: ICMP echo reply, id 678, seq 125, length 64
23:31:29.464637 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 678, seq 126, length 64
23:31:29.464676 IP 192.168.0.3 > 192.168.0.1: ICMP echo reply, id 678, seq 126, length 64
```

```
root@grave:~# tcpdump -i eth0
[ 3283.384989] device eth0 entered promiscuous mode
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
23:32:16.114032 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 678, seq 172, length 64
23:32:16.114067 IP 192.168.0.4 > 192.168.0.1: ICMP echo reply, id 678, seq 172, length 64
23:32:16.508993 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 678, seq 173, length 64
23:32:16.509041 IP 192.168.0.4 > 192.168.0.1: ICMP echo reply, id 678, seq 173, length 64
23:32:17.510594 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 678, seq 174, length 64
23:32:17.510634 IP 192.168.0.4 > 192.168.0.1: ICMP echo reply, id 678, seq 174, length 64
23:32:18.512267 IP 192.168.0.1 > 192.168.0.255: ICMP echo request, id 678, seq 175, length 64
23:32:18.512303 IP 192.168.0.4 > 192.168.0.1: ICMP echo reply, id 678, seq 175, length 64
```

Et on peut voir qu'à présent les autres machines répondent aux requêtes d'Immortal.

AVEC LA COMMANDE REBOOT, REDEMARREZ LA MACHINE IMMORTAL. VOUS NOTEZ QUE L'INTERFACE RESEAU A PERDU SA CONFIGURATION ! POUR REMEDIER A CE PROBLEME, IL FAUT EDITER LE FICHIER /ETC/NETWORK/INTERFACES ET Y DONNER LA CONFIGURATION DE L'INTERFACE ETH0 : IFACE ETH0 INET STATIC. CHERCHEZ SUR INTERNET ET DANS LE MAN (MAN INTERFACES) COMMENT CONFIGURER CE FICHIER. N'OUBLIEZ PAS DE METTRE UNE LIGNE AUTO ETH0 NOTEZ QUE CE FICHIER EST INTERPRETE SEULEMENT AU DEMARRAGE DE LA MACHINE, OU LORSQUE VOUS APPELEZ EXPLICTEMENT LE SCRIPT /ETC/INIT.D/NETWORKING RESTART. TESTEZ VOTRE CONFIGURATION.

Après le reboot :

```
root@immortal:~# ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1  (Local Loopback)
        RX packets 24  bytes 1632 (1.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 24  bytes 1632 (1.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

J'ai donc modifié le fichier interfaces :

```
root@immortal:~# cat /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto eth0
iface eth0 inet static
    address 192.168.0.1/24
```

Et ainsi après avoir exécuté le script /etc/init.d/networking restart, et ensuite avoir reboot immortal, on peut voir que la configuration est cette fois-ci bien restée :

```
root@immortal:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.1  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a8aa:aaff:feaa:0  prefixlen 64  scopeid 0x20<link>
        ether aa:aa:aa:aa:00:00  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 648 (648.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1  (Local Loopback)
        RX packets 24  bytes 1632 (1.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 24  bytes 1632 (1.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```