

Configuration du réseau sur un système d'exploitation GNU-Linux

Quelques définitions

Nous avons trois grands axes pour configurer un réseau.

L'adresse IP

Tout d'abord, une adresse IP est unique sur un réseau. C'est une suite de 4 nombres allant de 0 à 255 (inclus) séparés par des points (par exemple 192.168.1.32). Cette adresse IP appartient à une classe réseau. Il existe 5 classes d'IP :

- Classe A -> de 0.0.0.0 à 127.255.255.255
- Classe B -> de 128.0.0.0 à 191.255.255.255
- Classe C -> de 192.0.0.0 à 223.255.255.255
- Classe D -> Réservée à un usage multicast.
- Classe E -> Utilisée à titre expérimental.

De l'adresse IP et de sa classe, on peut en déduire trois grandes adresses:

- Le masque réseau (netmask) qui serait dans notre cas 255.255.255.0 (classe C) sert à identifier le réseau associé à une adresse IP.
- L'adresse réseau (network address) : 192.168.1.0, cette adresse ne peut être attribuée à aucun ordinateur sur le réseau.
- L'adresse de diffusion (broadcast address) : permettant d'envoyer un message à toutes les machines situées sur le réseau (ici 192.168.1.255).

Il existe la notation **CIDR**(Classless Inter-Domain Routing) (/24). Sous Linux, on a l'outil **ipcalc** qui permet de voir cette notation.

Exemple :

```
ipcalc 192.168.30.0/24
ipcalc 192.168.30.0/16   (classe B)
ipcalc 192.168.30.0/8   (Classe A)
```

La passerelle

La passerelle ou **GATEWAY** permet de relier deux réseaux informatiques différents. Elle est du type, dans notre cas : 192.168.1.1.

Le serveur DNS

Le serveur **DNS** ou **Domain Name System** établit la liaison entre un nom de domaine et une adresse IP. Grâce au serveur DNS nous pouvons retrouver une machine sur un réseau via son adresse IP ou bien sous sa forme canonique (par exemple www.domaine.fr).

Les fichiers de configuration

/etc/network/interfaces

Sur la distribution Debian, les paramètres réseaux vont être stockés à cet emplacement: **/etc/network/interfaces**. Editons ce fichier avec **vim**. Nous retrouvons l'adresse Loopback (127.0.0.1) autrement appelé le localhost qui boucle sur notre propre machine.

Nous allons maintenant attribuer les adresses nécessaires au bon fonctionnement de notre serveur, car celles-ci avaient été attribuées automatiquement lors de l'installation. Nous allons donc enlever la configuration DHCP en nous attribuant une adresse spécifique pour notre serveur (192.168.30.219), ce qui nous donne :

AVANT :

```
allow-hotplug eth0
iface eth0 inet dhcp
```

APRES :

```
auto eth0
iface eth0 inet static
    address 192.168.30.219
    netmask 255.255.255.0
    network 192.168.30.0
    broadcast 192.168.30.255
    gateway 192.168.30.1
```

Il faut maintenant relancer le fichier et vérifier que la configuration a bien été prise en compte.

```
# /etc/init.d/networking restart
# ifconfig
```

/etc/resolv.conf

Ce fichier contient les adresses IP des serveurs de noms :

```
$ cat /etc/resolv.conf
search mondomaine.fr autredomaine.fr
nameserver 192.168.30.1
```

Les options **search** et **domain** permettent de spécifier un nom de domaine à rajouter à toute requête portant sur un nom qui ne contient pas de point. Ce qui signifie ici qu'une recherche sur pc235 sera tentée en utilisant pc235.mondomaine.fr. Si aucune correspondance n'est trouvée, une autre recherche sera alors tentée avec pc235.autredomaine.fr, etc.

```
$ ping pc235
PING pc235.mondomaine.fr (192.168.30.235) 56(84) bytes of data.
...
```

L'option **domain** permet d'utiliser un nom (mais un seul) comprenant éventuellement un sous-domaine:

```
$ cat /etc/resolv.conf
domain truc.mondomaine.fr
nameserver 192.168.30.1
```

Dans ce cas, une recherche sera d'abord tentée avec **pc235.truc.mondomaine.fr**, puis, en cas d'échec, avec **pc235.mondomaine.fr**.

Ces deux options ne doivent pas figurer en même temps dans le fichier *resolv.conf*.

L'option **nameserver** permet d'indiquer l'adresse IP du serveur de noms. On peut mettre plusieurs lignes **nameserver** pour indiquer plusieurs serveurs de noms, mais attention, l'ordre à une importance : lors de la résolution d'un nom, le système va contacter le premier serveur DNS, et si il n'a pas obtenu de réponse au bout d'un certain temps (timeout généralement de 30s), il contactera le deuxième.

/etc/hostname

Ce fichier contient le nom de la machine (hostname). Il est utilisé lors du démarrage de la machine pour positionner le nom de la machine.

/etc/hosts

Le fichier **/etc/hosts** est l'ancêtre du DNS. A la création de l'Internet, il y avait très peu d'ordinateurs connectés, et ce fichier contenait la liste de ces ordinateurs et de leurs adresses IP. Chaque fois que de nouveaux ordinateurs étaient connectés à Internet, il fallait mettre à jour ce fichier pour accéder à ces nouveaux ordinateurs. Vu le nombre croissant d'ordinateurs connectés à Internet, ce fonctionnement n'était plus adapté et a donné lieu à l'invention du DNS.

Désormais, ce fichier permet de connaître les adresses IP d'ordinateurs non présents dans le DNS.

Ce fichier contient une ligne pour l'entrée **loopback** :

```
127.0.0.1      localhost
```

Ce fichier doit aussi contenir une ligne contenant le nom de l'ordinateur et son adresse IP; ceci est utile en cas de défaillance du serveur DNS :

```
192.168.30.230 pc230.mondomaine.fr pc230 pcalex
```

A la fin de ce fichier, on trouve désormais des entrées pour IPv6 :

```
::1          ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
ff02::3     ip6-allhosts
```

/etc/host.conf

Historiquement, ce fichier contenait l'ordre de recherche pour la résolution des noms : d'abord la consultation du fichier **/etc/hosts**, et ensuite l'interrogation du DNS :

```
order hosts,bind
```

Désormais, ces préférences d'ordre de recherche sont indiqués dans le fichier **/etc/nsswitch.conf** (voir ci-dessous).

/etc/nsswitch.conf

Ce fichier remplace désormais le fichier **/etc/host.conf** et permet d'indiquer au système l'ordre de recherche pour la résolution des noms :

```
more /etc/nsswitch.conf
...
hosts:          files dns
networks:       files
...
```

L'option **files** indique au système de consulter d'abord les fichiers (**/etc/hosts** pour les noms d'hotes et **/etc/networks** pour les noms de domaines) avant d'aller interroger le DNS.

/etc/networks

Ce fichier contient le nom des réseaux et leur adresse réseau :

```
default        0.0.0.0
loopback       127.0.0.0
link-local     169.254.0.0
mondomaine.fr  192.168.30.0
```

Les commandes

hostname

La commande **hostname** permet de connaître le nom de l'ordinateur :

```
# hostname
pc230
```

L'option **-f** (**-f** ou **--fqdn**) permet de connaître le nom de l'ordinateur avec son nom de domaine (**F**ull **Q**ualified **D**omain **N**ame) :

```
# hostname -f
pc230.mondomaine.fr
```

La commande **hostname** permet aussi de changer dynamiquement nom de machine, mais attention, cette modification est temporaire et ne sera plus active après un reboot. Pour changer de manière permanente le nom de la machine, il faut modifier le fichier **/etc/hostname** et le fichier **/etc/hosts** (voir ci-dessus) :

```
# hostname
pc230
# hostname pcalex
# hostname
pcalex
```

ifconfig

Utilisé sans argument, la commande **ifconfig** affiche les interfaces réseaux actives :

```
# ifconfig
eth0    Lien encap:Ethernet  HWaddr 00:05:5D:E1:F6:11
        inet adr:192.168.30.230  Bcast:192.168.30.255  Masque:255.255.255.0
        adr inet6: fe80::205:5dff:fe01:f611/64  Scope:Lien
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:16383 errors:0 dropped:0 overruns:0 frame:0
        TX packets:5998 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 lg file transmission:1000
        RX bytes:8467107 (8.0 MiB)  TX bytes:810927 (791.9 KiB)
        Interruption:10 Adresse de base:0xc800

lo      Lien encap:Boucle locale
        inet adr:127.0.0.1  Masque:255.0.0.0
        adr inet6: ::1/128  Scope:Hôte
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:24 errors:0 dropped:0 overruns:0 frame:0
        TX packets:24 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 lg file transmission:0
        RX bytes:2352 (2.2 KiB)  TX bytes:2352 (2.2 KiB)
```

L'option **-a** permet d'afficher toutes les interfaces, y compris celles qui ne sont pas activées.

On peut aussi spécifier à **ifconfig** l'interface à afficher :

```
# ifconfig eth0
...
```

La commande **ifconfig** permet également de configurer une interface réseau. Attention, cette modification est faite dynamiquement et ne sera plus active après un reboot.

```
# ifconfig eth0 10.0.0.1 netmask 255.0.0.0
```

La commande **ifconfig** peut aussi créer des alias d'interfaces réseaux. On peut ainsi affecter plusieurs adresses à une seule interface réseaux.

```
# ifconfig eth0:0 192.168.0.100
```

```
# ifconfig
# eth0    Link encap:Ethernet  HWaddr 00:1e:8c:26:af:c5
        UP BROADCAST MULTICAST  MTU:1500  Metric:1
        Packets reçus:0 erreurs:0 :0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 lg file transmission:1000
        Octets reçus:0 (0.0 B) Octets transmis:0 (0.0 B)

eth0:0   Link encap:Ethernet  HWaddr 00:1e:8c:26:af:c5
        inet adr:192.168.0.100  Bcast:192.168.0.255  Masque:255.255.255.0
        UP BROADCAST MULTICAST  MTU:1500  Metric:1
```

Attention : Si vous supprimez l' interface principale (ici : eth0), tous les alias qui en dépendent seront supprimés.

À faire...

parler de ifconfig up et down

arp

Le [protocole ARP](#) [\[archive\]](#) permet de trouver l'adresse MAC d'un ordinateur de mon réseau en fonction de son adresse IP.

La commande **arp** permet d'afficher la table de correspondance **adresses IP => adresses MAC** :

```
# arp -a
fw.mondomaine.fr (192.168.30.1) at 00:10:5A:DC:2B:4B [ether] on eth0
pc235.mondomaine.fr (192.168.30.235) at 00:11:95:DD:FD:F3 on eth0
```

Attention, cette table de correspondance est régulièrement vidée.

route

La commande **route** permet d'afficher la table de routage réseau :

```
# route
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic Metric Ref      Use Iface
dl2.mondomaine.fr *          255.255.255.0   U      0      0      0 eth0
default          fw.mondomaine.fr 0.0.0.0         UG     0      0      0 eth0
```

L'option **-n** affiche la table de routage réseau sans remplacer les adresses IP par leurs noms canoniques :

```
# route -n
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic Metric Ref      Use Iface
192.168.30.0     0.0.0.0         255.255.255.0   U      0      0      0 eth0
0.0.0.0          192.168.30.1   0.0.0.0         UG     0      0      0 eth0
```

On peut obtenir le même résultat en utilisant la commande **netstat** :

```
# netstat -rn
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic  MSS Fenêtre irtt Iface
192.168.30.0     0.0.0.0         255.255.255.0   U      0  0      0 eth0
0.0.0.0          192.168.30.1   0.0.0.0         UG     0  0      0 eth0
```

La commande **route** permet de modifier la table de routage. On peut ainsi rajouter une route pour contacter un réseau ou une machine particulière.

Exemple : joindre le réseau 10.0.0.0/8

```
pc210:~# route -n
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic Metric Ref      Use Iface
192.168.30.0     0.0.0.0         255.255.255.0   U      0      0      0 eth0
0.0.0.0          192.168.30.1   0.0.0.0         UG     0      0      0 eth0
pc210:~#
pc210:~# route add -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.30.99 dev eth0
pc210:~#
```

```
pc210:~# route -n
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic Metric Ref      Use Iface
192.168.30.0    0.0.0.0         255.255.255.0   U      0      0        0 eth0
10.0.0.0      192.168.30.99 255.0.0.0      UG    0    0        0 eth0
0.0.0.0         192.168.30.1   0.0.0.0         UG     0      0        0 eth0
```

Exemple : joindre la machine 1.2.3.4 :

```
pc210:~# route -n
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic Metric Ref      Use Iface
192.168.30.0    0.0.0.0         255.255.255.0   U      0      0        0 eth0
0.0.0.0         192.168.30.1   0.0.0.0         UG     0      0        0 eth0
pc210:~#
pc210:~# route add -host 1.2.3.4 gw 192.168.30.98 dev eth0
pc210:~#
pc210:~# route -n
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic Metric Ref      Use Iface
1.2.3.4      192.168.30.98 255.255.255.255 UGH    0    0        0 eth0
192.168.30.0    0.0.0.0         255.255.255.0   U      0      0        0 eth0
0.0.0.0         192.168.30.1   0.0.0.0         UG     0      0        0 eth0
```

À faire...

parler de comment rendre une route permanente (modif de /etc/network/interfaces)

ping

La commande **ping** permet de tester si on arrive à joindre un serveur donné. Cette commande utilise le protocole **ICMP** et envoi des packets ICMP de type **echo-request**. Si l'ordinateur distant est joignable, il répondra à ce message par un packet ICMP **echo-reply**.

```
$ ping google.fr
PING google.fr (216.239.59.104) 56(84) bytes of data.
64 bytes from 216.239.59.104: icmp_seq=1 ttl=236 time=43.2 ms
64 bytes from 216.239.59.104: icmp_seq=2 ttl=236 time=44.3 ms
64 bytes from 216.239.59.104: icmp_seq=3 ttl=236 time=43.1 ms
64 bytes from 216.239.59.104: icmp_seq=4 ttl=236 time=72.2 ms
...
CTRL-c pour arreter
```

```
$ ping -c 2 google.fr
PING google.fr (216.239.59.104) 56(84) bytes of data.
64 bytes from 216.239.59.104: icmp_seq=1 ttl=236 time=43.6 ms
64 bytes from 216.239.59.104: icmp_seq=2 ttl=236 time=44.6 ms
```

```
--- google.fr ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 5104ms
rtt min/avg/max/mdev = 43.643/44.164/44.685/0.521 ms
```

Si l'ordinateur est injoignable, on n'obtiendra pas de réponse à nos packets ICMP :

```
$ ping pc235
PING pc235.mondomaine.fr (192.168.30.235) 56(84) bytes of data.
From pc230.mondomaine.fr (192.168.30.230) icmp_seq=2 Destination Host Unreachable
From pc230.mondomaine.fr (192.168.30.230) icmp_seq=3 Destination Host Unreachable
```

Attention, si l'ordinateur distant ne répond pas, cela ne signifie pas obligatoirement qu'il est indisponible. Le problème peut venir du réseau (la commande **traceroute** permettra de déterminer l'endroit qui bloque), ou il peut être paramétré pour ne pas répondre au protocole ICMP, ou un firewall sur la route peut bloquer le protocole ICMP.

L'option **-c** de la commande **ping** permet d'indiquer le nombre de packets à envoyer :

```
$ ping -c 2 google.fr
PING google.fr (216.239.59.104) 56(84) bytes of data.
64 bytes from 216.239.59.104: icmp_seq=1 ttl=236 time=43.6 ms
64 bytes from 216.239.59.104: icmp_seq=2 ttl=236 time=44.6 ms
```

```
--- google.fr ping statistics --- 2 packets transmitted, 2 received, 0% packet loss, time 5104ms rtt
min/avg/max/mdev = 43.643/44.164/44.685/0.521 ms
```

traceroute

La commande **traceroute** permet d'afficher la route empruntée pour atteindre un ordinateur donné :

```
# traceroute www.google.fr
traceroute: Warning: www.google.fr has multiple addresses; using 209.85.135.103
traceroute to www.l.google.com (209.85.135.103), 30 hops max, 40 byte packets
 1 fw.mondomaine.fr (192.168.30.1)  1.854 ms  0.323 ms  0.281 ms
 2 192.168.10.1 (192.168.10.1)  0.658 ms  0.594 ms  0.522 ms
 ...
```

mtr

La commande **mtr** (**my traceroute**) permet aussi d'afficher la route empruntée pour atteindre un ordinateur donné, mais ré-actualise la liste en permanence :

```
pc230 (0.0.0.0)          My traceroute  [v0.71]          Tue Dec  4 13:13:23 2007

Host                    Packets  Loss%  Snt  Last  Avg  Best  Wrst  StDev
1. fw.mondomaine.fr    0.0%    39    0.7  0.6  0.4  7.4  1.1
2. 192.168.10.1        0.0%    39    0.6  0.8  0.6  7.2  1.1
 ...
```


nslookup

La commande **nslookup** permet d'interroger un serveur DNS :

```
$ nslookup
> set type=any
> google.fr
Server:          192.168.30.1
Address:         192.168.30.1#53

Non-authoritative answer:
Name:   google.fr
Address: 216.239.59.104
Name:   google.fr
Address: 66.249.93.104
Name:   google.fr
Address: 72.14.221.104
google.fr      nameserver = ns4.google.com.
google.fr      nameserver = ns1.google.com.
google.fr      nameserver = ns2.google.com.
google.fr      nameserver = ns3.google.com.

Authoritative answers can be found from:
google.fr      nameserver = ns2.google.com.
google.fr      nameserver = ns3.google.com.
google.fr      nameserver = ns4.google.com.
google.fr      nameserver = ns1.google.com.
ns1.google.com internet address = 216.239.32.10
>
CTRL-d pour sortir
```

Cette commande est interactive, elle nécessite l'intervention de l'utilisateur. On utilise désormais les commandes **host** et **dig** pour effectuer les mêmes opérations (voir ci-dessous).

host

La commande **host** permet d'effectuer une résolution DNS :

```
$ host pc235
pc235.mondomaine.fr has address 192.168.30.235
pc235.mondomaine.fr mail is handled by 0 pc235.mondomaine.fr.
```

Elle permet aussi d'effectuer une résolution DNS inverse (quel nom canonique est associé à une adresse IP donnée ?) :

```
$ host 192.168.30.235
235.30.168.192.in-addr.arpa domain name pointer pc235.mondomaine.fr.
```

dig

La commande **dig** permet d'interroger les enregistrements DNS (appelé aussi **Ressources Records** ou **RR**) d'un nom de domaine donné :

```
$ dig google.fr
...
;; QUESTION SECTION:
;google.fr.                IN      A

;; ANSWER SECTION:
google.fr.                 205     IN      A      66.249.93.104
google.fr.                 205     IN      A      72.14.221.104
...
```

On peut indiquer à la commande **dig** le **Ressource Record** que l'on désire connaître :

```
$ dig google.fr MX
...
;; ANSWER SECTION:
google.fr.                 10800   IN      MX     10 smtp1.google.com.
google.fr.                 10800   IN      MX     10 smtp2.google.com.
google.fr.                 10800   IN      MX     10 smtp3.google.com.
...
```

On peut indiquer à la commande **dig** le serveur DNS à interroger :

```
$ dig @ns1.google.com google.fr
...
;; ANSWER SECTION:
google.fr.                 1800    IN      A      66.249.93.104
google.fr.                 1800    IN      A      216.239.59.104
...
```

whois

La commande **whois** permet d'interroger la base de données **whois** contenant les informations sur le propriétaire du domaine et les personnes responsables pour les aspects administratif et technique :

```
$ whois google.fr
...
domain:      google.fr
address:     GOOGLE INC
address:     28, rue Juliette Lamber
address:     75017 Paris
address:     FR
admin-c:     VB2334-FRNIC
tech-c:      NA25-FRNIC
zone-c:      NFC1-FRNIC
nserver:     ns1.google.com
...
```