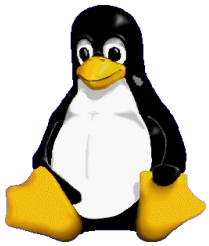

Le système GNU/Linux

By ShareVB

Réseau et configuration IP

Table des matières

I. Adresse IP et nom DNS

A chaque équipement réseau, un switch, une machine, un routeur, on peut associer une adresse IP. Mais on peut aussi associer un nom DNS à une IP plus simple à retenir. Il est plus simple de retenir www.google.fr que son équivalent numérique IP.

Cela permet de faire ping www.google.fr ou ping 216.239.59.104 pour faire un ping sur google.fr.

Il ne faut pas confondre le nom DNS qui est relatif au protocole TCP/IP et le nom Netbios donné sur les réseaux Microsoft.

Enfin, on associe une adresse IP à une adresse MAC Ethernet sur les réseaux Ethernet. Pour savoir quel adresse MAC a une IP d'une machine d'un réseau local, on utilise le protocole ARP. On stocke alors pour un petit moment les associations IP <-> adresses MAC afin de pouvoir faire la communication entre la machine locale et les machines dont on a une correspondance dans la table ARP.

Pour que la résolution de nom DNS fonctionne, il faut soit utiliser un serveur DNS qui contient les tables IP/Noms, soit déclarer dans un fichier, tous les noms d'hôtes que l'on veut utiliser, et pour chaque nom, son adresse IP. Cette déclaration est réalisée dans le fichier "`/etc/hosts`" (qui sous Windows se trouve dans C:\Windows\system32\drivers).

Il existe une interface particulière sur TCP/IP qui n'est pas une carte réseau réelle. C'est l'interface loopback qui permet à des programmes TCP/IP de communiquer entre eux sans sortir de la machine

locale. Cela permet aussi de tester si le module « réseau » du système fonctionne bien. On peut ainsi faire un `ping localhost`. Tous les paquets qui entre par localhost resortent par localhost. On récupère donc toujours les paquets que l'on envoie.

II. Nom de la machine locale sous Debian

On peut obtenir le nom DNS de la machine locale par la commande `hostname`. Ce nom d'hôte est stocké dans deux fichiers : `/etc/hosts` et `/proc/sys/kernel/hostname`.
Pour le changer, on fera :

- `hostname` *`nouveau_nom_hôte`*
- dans le fichier `/etc/hosts`

```
127.0.0.1     nouveau_nom_hôte localhost.localdomain localhost
```

- pour le fichier `/proc/sys/kernel/hostname`

```
[root]# echo « nouveau_nom_hôte » > /proc/sys/kernel/hostname
```

III. Nom de la machine locale sous Fedora

On peut obtenir le nom DNS de la machine locale par la commande `hostname`. Ce nom d'hôte est stocké dans deux fichiers : `/etc/hosts` et `/etc/sysconfig/network`.
Pour le changer, on fera :

- `hostname` *`nouveau_nom_hôte`*
- dans le fichier `/etc/hosts`

```
127.0.0.1     nouveau_nom_hôte localhost.localdomain localhost
```

- dans le fichier `/etc/sysconfig/network`

```
NETWORKING=yes
HOSTNAME=nouveau_nom_hôte
```

IV. Résolution des noms et DNS

1 Statique

Dans le fichier `/etc/hosts`, on peut ajouter des nom DNS pour des IP connues et fixes. Par exemple :

```
192.168.34.56  unserveur
192.168.34.1   monpc
```

On pourra alors taper `ping unserveur` à la place de `ping 192.168.34.56`

2 Dynamique par DNS

Pour obtenir le nom DNS d'une IP et inversement, on pourra exécuter au choix :

```
host <ip> ou host <nom>
nslookup <ip> ou nslookup <nom>
dig -x <ip> ou dig <nom>
```

Il est nécessaire de définir un ou deux serveurs DNS par défaut du système afin que celui-ci puisse faire la résolution des noms DNS automatiquement pour toutes les applications. Pour cela, il suffit d'éditer le fichier `/etc/resolv.conf` et d'y inscrire :

```
nameserver ip_du_server_DNS_primaire
nameserver ip_du_server_DNS_secondaire
```

On peut aussi spécifier dans ce fichier, le nom de domaine local (directive `domain <nom du domaine local>`) et une liste de noms de domaines à ajouter en cas de résolution infructueuse d'un nom DNS sans domaine (directive `search <nom domaine local> <autre nom domaine> ...`). Par exemple si on a `search sharevb.net` et que l'on exécute `dig www`, on va trouver l'IP de www.sharevb.net.

On peut spécifier l'ordre de recherche fichier hosts puis DNS dans le fichier `/etc/host.conf` :

```
order hosts,bind
```

V. Configuration des interfaces sous Debian

Les interfaces permettent d'accéder aux réseaux TCP/IP. Les IP affectées aux différentes interfaces (cartes réseaux, aliasing IP...) peuvent être définies au démarrage dans le fichier `/etc/network/interfaces` pour chaque interface :

```
#démarrer l'interface dès le démarrage
auto nom_interface
ifce nom_interface inet static
     address <IP de l'interface>
     netmask <masque du réseau de l'interface>
     gateway <IP de la passerelle pour l'interface>

#interface avec DHCP
ifce nom_interface inet dhcp
```

Pour changer dynamiquement la configuration IP d'une interface :

```
     [root]# ifconfig <interface> <nouvelle adresse IP> [netmask <masque de sous réseau>]
```

La partie entre [] est facultative.

Cette commande permet aussi d'obtenir l'adresse IP d'une interface ainsi que le nombre de paquets reçus et transmis. Pour cela, il suffit de l'appeler sans argument (infos sur toutes les interfaces) ou avec le nom de l'interface dont on veut des informations.

VI. Configuration des interfaces sous Fedora

Les interfaces permettent d'accéder aux réseaux TCP/IP. Les IP affectées aux différentes interfaces (cartes réseaux, aliasing IP...) peuvent être définies au démarrage dans les fichiers `/etc/sysconfig/network-scripts/ifcfg-<nom interface>` (par exemple, ifcfg-eth0) :

```
DEVICE=<nom interface>
# static : IP fixe pour l'interface
# none : idem
# dhcp : utiliser DHCP pour obtenir une IP au démarrage
# bootp : utiliser BOOTP pour obtenir une IP au démarrage
BOOTPROTO=<protocole>
#adresse MAC de la carte réseau de l'interface
HWADDR=XX:XX:XX:XX:XX:XX
#configuration du sous réseau auquel l'interface est connectée
IPADDR=<adresse IP de la machine>
NETMASK=<masque du sous réseau>
NETWORK=<adresse du sous réseau>
BROADCAST=<adresse broadcast du sous réseau>
# activer au démarrage (yes) ou pas (no)
ONBOOT=yes
```

Pour changer dynamiquement la configuration IP d'une interface :

```
[root]# ifconfig <interface> <nouvelle adresse IP> [netmask <masque de sous réseau>]
```

La partie entre [] est facultative.

Cette commande permet aussi d'obtenir l'adresse IP d'une interface ainsi que le nombre de paquets reçus et transmis. Pour cela, il suffit de l'appeler sans argument (infos sur toutes les interfaces) ou avec le nom de l'interface dont on veut des informations.

VII. Table de routage et commande route

La table de routage permet d'indiquer et de savoir rapidement par quelle interface ou/et passerelle doivent sortir les paquets pour atteindre une adresse IP ou un réseau précis.

La commande route permet de modifier cette table :

- route par une interface de la machine (physique ou virtuelle) vers une machine ou un réseau

```
route add <IP machine> dev <nom d'interface>
route add -net <IP réseau> netmask <masque réseau> dev <nom d'interface>
```

- route par une passerelle vers une machine ou un réseau

```
route add <IP machine> gateway <IP passerelle>
route add -net <IP réseau> netmask <masque réseau> gateway <IP passerelle>
```

- route par défaut (par une passerelle)

```
route add default gateway <IP passerelle>
```

Il est à noter que `default` vaut en réalité `0.0.0.0 netmask 255.255.255.255`.

Il est aussi à noter que lors de l'appel à ifconfig, la table de routage est modifiée :

```
[root]# ifconfig eth0 192.168.34.1
```

peut se développer en

```
[root]# ifconfig eth0 192.168.34.1 netmask 255.255.255.0 up
[root]# route add -net 192.168.34.0 netmask 255.255.255.0 dev eth0
```

VIII. Interfaces virtuelles

Les interfaces virtuelles permettent d'ajouter plusieurs IP sur la même interface physique.

On peut ajouter dynamiquement plusieurs IP sur la même interface avec la commande ifconfig :

```
[root]# ifconfig eth0 <IP principale>
[root]# ifconfig eth0:0 <IP secondaire>
...
[root]# ifconfig eth0:n <n-ième IP>
```

Cela peut servir à faire sortir le trafic vers Internet par une interface et le traffic vers le réseau local sur une autre interface sans avoir plusieurs LAN et cartes réseaux. Cela peut aussi servir à se connecter sur un réseau LAN1 et LAN2 (reliés par le même switch) sans avoir plusieurs cartes réseaux.

Par exemple, si on a un réseau local 172.20.0.0/16 et que l'on veut accéder à un sous réseau 172.20.9.0/24 par une autre interface, on ferra :

```
[root]# ifconfig eth0 172.20.0.0 netmask 255.255.0.0 up
[root]# ifconfig eth0:0 172.20.9.0 netmask 255.255.255.0 up
```

1 Sous Debian

Pour activer plusieurs interfaces sur la même carte réseau au démarrage, on configure ces interfaces dans le fichier `/etc/network/interfaces` pour chaque interface/interface virtuelle. Dans notre exemple :

```
#démarrer l'interface dès le démarrage
auto nom_interface
ifce nom_interface inet static
     address 172.20.9.x
     netmask 255.255.0.0
     gateway 172.168.0.10

#interface virtuelle
auto eth0:0
ifce eth0:0 inet static
     address  172.20.9.(100+x)
     netmask  255.255.255.0
```

2 Sous Fedora

Pour activer plusieurs interfaces sur la même carte réseau au démarrage, il suffit de mettre la configuration dans les fichiers `/etc/sysconfig/network-scripts/ifcfg-eth0`, `ifcfg-eth0:0`...Dans notre exemple :

Configurer l'interface virtuelle eth0:0 `(ifcfg-eth0:0)` :

```
BOOTPROTO=none      #surtout pas DHCP
DEVICE=eth0:0       #nom d'interface
NETMASK=255.255.0.0 #masque du sous réseau
IPADDR=172.20.9.x #adresse IP
ONBOOT=yes          #active au démarrage
```

Configurer l'interface principale eth0 `(ifcfg-eth0)` :

```
BOOTPROTO=none      #ou dhcp
```

```
DEVICE=eth0:0        #nom d'interface
NETMASK=255.255.255.0 #masque du sous réseau
IPADDR=172.20.9.x      #adresse IP
ONBOOT=yes           #active au démarrage
```

IX. Table ARP

La table arp contient des associations adresse IP <-> adresse MAC. En effet, les cartes réseaux et réseaux Ethernet ne savent par dialoguer directement avec des IP. Il leur est nécessaire de connaitre les adresses MAC des cartes du réseaux afin de s'échanger les paquets IP.

On peut consulter le contenu de la table arp avec la commande `arp -a` ou `arp -na` suivant que l'on souhaite la résolution des noms DNS ou pas.