

Configuration du réseau

1. Configuration universelle du réseau

Chaque distribution Linux essaye de faire en sorte que les paramètres réseau soient aussi faciles que possible à configurer. Le but est souvent de ne pas souffrir de la comparaison avec Windows, et de faire en sorte que l'utilisateur ait à sa disposition une interface intuitive et facile à configurer. Cette configuration se fait avec des utilitaires, graphiques ou non, et des fichiers de configuration que liront les scripts de lancement du réseau.

Indépendamment de ces éléments de confort apportés par les distributions ou les bureaux graphiques, on aura toujours, quelle que soit la distribution et l'environnement, les commandes de base permettant la configuration du réseau, à savoir l'adresse ip, la route par défaut, et l'adresse des serveurs DNS. La démarche indiquée ci-après, si elle n'est pas la plus rapide (quoique), a l'avantage de l'universalité.

a. Détermination de l'interface réseau

Les systèmes Linux utilisent un nom symbolique par interface réseau, qu'il s'agisse d'une interface réelle ou virtuelle, ethernet ou autre. Dans le cas courant où le système est connecté à un réseau ethernet et n'utilise qu'une seule carte, cette carte sera désignée « eth0 ». On pourra déterminer la liste de toutes les interfaces réseaux existant sur un système, configurée ou non par la commande **ifconfig**.

Détermination des interfaces réseau par la commande ifconfig

```
ifconfig -a
```

b. Affectation de l'adresse IP : ifconfig

La commande **ifconfig** a de nombreux usages, et elle est surtout connue pour afficher les adresses MAC et IP pour un système déjà configuré. Néanmoins, la commande **ifconfig** peut aussi être utilisée pour affecter dynamiquement l'adresse et le masque d'une machine.

Affectation d'une adresse IP avec la commande ifconfig

```
ifconfig interface adresse_ip  
ifconfig interface netmask masque
```

Même si ça n'est pas le plus courant des usages, il est possible d'ajouter une deuxième adresse IP à une interface déjà configurée.

Ajout d'une adresse IP secondaire à une interface

```
ifconfig interface:sous-interface adresse_ip
```

Commande ifconfig : options et paramètres	
<i>interface</i>	Nom Linux de l'interface. Par exemple eth0.
<i>sous-interface</i>	Nom arbitraire de la sous-interface. Chaîne de caractères quelconque.
<i>adresse_ip</i>	Adresse IP à affecter à la machine.
<i>masque</i>	Valeur du masque de sous-réseau associé à l'adresse IP.

c. Configuration du client DNS : fichier /etc/resolv.conf

Les machines Linux disposent nativement d'un client DNS appelé **resolver**. Toute application fonctionnant sur Linux et ayant besoin de faire une requête DNS s'appuiera sur ce composant.


Il exploite le fichier de configuration simple **/etc/resolv.conf** où doit se trouver la référence d'au moins un serveur

DNS.

Format simplifié du fichier /etc/resolv.conf

```
search domaine
nameserver adresse_ip
```

Fichier /etc/resolv.conf : directives et variables utilisées	
search	Facultatif : indique le suffixe de recherche employé sur le poste Linux. Permet de ne pas taper l'intégralité du nom de domaine pleinement qualifié (FQDN) dans les applications. Le fichier /etc/resolv.conf admet plusieurs domaines de recherches précisés par search.
domaine	Le FQDN du domaine constituant le suffixe de recherche.
nameserver	Indique l'adresse IP du serveur DNS qui assurera les résolutions. Le fichier /etc/resolv.conf admet plusieurs serveurs DNS précisés par nameserver.
adresse_ip	Adresse IP du serveur DNS à interroger.

 Certaines documentations préconisent l'usage de la commande `hostname -d` pour connaître le suffixe DNS d'un système. Il s'agit du suffixe attaché au nom d'hôte, et non au client DNS. Il n'est donc pas consulté lors des résolutions DNS.

d. Configuration de la passerelle par défaut : route

La commande **route** permet de définir des routes statiques sur une machine Linux. Dans le cadre d'une configuration simple et ponctuelle, on pourra l'utiliser pour définir la passerelle par défaut. Il s'agira en fait de déclarer une route statique indiquant la route par défaut.

Syntaxe de la commande route pour indiquer une route statique

```
route add -net réseau_dest netmask masque gw ip_passerelle
```

Syntaxe de la commande route pour indiquer la passerelle par défaut

```
route add -net 0.0.0.0 gw ip_passerelle
route add default gw ip_passerelle
```

Commande route : options et paramètres	
add	Indique que l'on ajoute une route à la table de routage.
-net	Indique que la destination est un réseau.
réseau_dest	Le réseau à atteindre par la route statique qu'on paramètre.
0.0.0.0	La route par défaut. 0.0.0.0 représente tous les réseaux possibles.
gw	Annonce la valeur de la passerelle.
ip_passerelle	Adresse IP de la passerelle à utiliser.
default	Équivalent à -net 0.0.0.0
masque	Le masque de sous-réseau associé à la route ajoutée.

Un serveur Linux utilisé en tant que routeur supporte aussi les principaux protocoles de routage. Le logiciel

historique **routed** supporte uniquement le protocole de routage RIP, alors que la suite logicielle plus moderne **quagga** permet d'exploiter la quasi-totalité des protocoles de routages IP.

e. Configuration du nom d'hôte : **hostname**

Le nom d'hôte de la machine peut être affecté dynamiquement avec la commande **hostname**. Il permet aussi d'afficher le nom d'hôte du système s'il est appelé sans argument.

Syntaxe de la commande **hostname** pour affecter un nom d'hôte

```
hostname nom_hote
```

nom_hote représentant le nom qu'on souhaite affecter au système.

Attention, cette valeur est conservée en mémoire vive, et sera perdue dès que le système redémarrera. Les systèmes ordinaires en production doivent donc conserver cette valeur dans un fichier de configuration qui est lu à chaque démarrage. Ce fichier dépend de la distribution. C'est par exemple **/etc/hostname** pour les distributions d'origine Debian, et **/etc/sysconfig/network** pour les distributions d'origine RedHat. Les scripts exécutés au démarrage du système se chargent d'appeler la commande **hostname** et récupèrent la valeur du nom du système dans le fichier.

Exemple de contenu d'un fichier **/etc/hostname**

```
root@firmin:~$ cat /etc/hostname
firmin
```

2. Spécificité des distributions

Les seules règles universelles pour la configuration du réseau sont celles décrites dans les paragraphes précédents. Les distributions Linux courantes ont néanmoins des procédures de configuration par scripts et fichiers de configuration qu'on peut classer en deux grandes familles : celles dont la configuration réseau est située dans le répertoire **/etc/network**, et celles dont la configuration réseau est située dans le répertoire **/etc/sysconfig/network-scripts**.

a. Configuration réseau dans **/etc/network**

C'est le cas des distributions Debian et dérivées. Les éléments de configuration sont situés dans un fichier au format simple : **/etc/interfaces**.

Format du fichier de configuration **/etc.network/interfaces** pour une adresse IP statique

```
auto interface
iface interface inet static
address adresse_ip
netmask masque
gateway ip_passerelle
```

Format du fichier de configuration **/etc.network/interfaces** pour une adresse IP dynamique

```
auto interface
iface interface inet dhcp
```

Fichier interfaces : options et paramètres	
auto	Indique que l'interface devra être activée automatiquement au démarrage.
interface	Le nom linuxien de l'interface à configurer (exemple : eth0).
inet	Indique qu'on va affecter une adresse Ipv4.
static	Indique que l'adresse IP configurée sera statique.

<i>adresse_ip</i>	Adresse IP à affecter à l'interface.
<i>masque</i>	Masque de sous-réseau à affecter à l'interface.
<i>ip_passerelle</i>	Adresse IP de la passerelle par défaut.
dhcp	Indique que l'adresse IP configurée sera dynamique et obtenue par requête DHCP.

Ces fichiers n'ont évidemment aucune action en eux-mêmes, ils sont appelés par le script de lancement du service réseau (en général /etc/init.d/networking), lequel script invoquera la commande **ifup** (interface up) pour activer les interfaces avec leurs paramètres réseau.

b. Configuration réseau dans /etc/sysconfig/network-scripts

C'est le cas des distributions RedHat et dérivées. Les éléments de configuration sont situés dans un fichier au format simple par interface situé dans le répertoire /etc/sysconfig/network-scripts. Ces fichiers ont tous le préfixe **ifcfg-** suivi du nom de l'interface à configurer.

Format du fichier ifcfg-interface pour une adresse IP statique

```
DEVICE=interface
BOOTPROTO=none
ONBOOT=yes
IPADDR=adresse_ip
NETMASK=masque
GATEWAY=ip_passerelle
```

Format du fichier ifcfg-interface pour une adresse IP dynamique

```
DEVICE=interface
BOOTPROTO=dhcp
ONBOOT=yes
```

Fichier ifcfg : options et paramètres	
interface	Le nom Linux de l'interface à configurer (exemple : eth0).
BOOTPROTO=dhcp	Indique que l'adresse IP configurée sera dynamique et obtenue par requête DHCP.
ONBOOT=yes	Indique que l'interface devra être activée automatiquement au démarrage.
<i>adresse_ip</i>	Adresse IP à affecter à l'interface.
<i>masque</i>	Masque de sous-réseau à affecter à l'interface.
<i>ip_passerelle</i>	Adresse IP de la passerelle par défaut.



Quel que soit le format des fichiers de configuration réseau, le paramètre précisant l'adresse de passerelle à proximité de la configuration d'une interface pourrait faire penser que la passerelle est attachée à l'interface. Or une passerelle par défaut, quel que soit le système, est unique et liée à la table de routage du système et non à une quelconque interface.

3. Autres commandes et fichiers de gestion du réseau

Nous avons vu que les paramètres réseau pouvaient être configurés avec les seules commandes **ifconfig** et **route**. Il existe néanmoins de nombreux autres utilitaires qui permettent d'administrer, configurer et diagnostiquer le fonctionnement du réseau.

a. Gestion des adresses MAC avec arp

Tout équipement réseau qui exploite le protocole IP sur un réseau ethernet est tenu, pour établir la correspondance entre les adresses IP et les adresses MAC, d'utiliser le protocole ARP. Dans un fonctionnement dynamique, cas le plus courant, une machine connaissant l'adresse IP de son destinataire mais ayant besoin de renseigner son en-tête MAC pour communiquer envoie un broadcast pour demander si quelqu'un sur le réseau possède l'adresse IP en question. Si la machine destinataire est à portée de broadcast (c'est-à-dire dans le réseau local), elle répond en unicast et indique son adresse MAC. La résolution ARP est alors réalisée. Les correspondances établies entre adresses MAC et adresses IP sont conservées un certain temps en mémoire dans ce qu'on appelle le cache ARP. Dans quelques cas particuliers, on peut aussi affecter de façon statique une correspondance entre adresse IP et adresse MAC.

La commande **arp** permet d'observer et éventuellement de gérer les valeurs contenues dans ce cache.

Syntaxe de la commande arp pour observer le cache

```
arp -n
```

Le paramètre -n n'est pas obligatoire, mais il dispense le système de réaliser une recherche DNS inverse qui ralentit énormément l'affichage.

Syntaxe de la commande arp pour effacer une entrée du cache

```
arp -d adresse_ip
```

Syntaxe de la commande arp pour affecter une valeur au cache

```
arp -s adresse_ip adresse_mac
```

Où *adresse_ip* représente l'adresse IP de l'entrée que l'on souhaite gérer, et *adresse_mac* représente l'adresse MAC d'une entrée à associer à une adresse IP. Les adresses MAC sont exprimées sous forme d'octets en hexadécimal séparés par des doubles points.

L'usage courant est naturellement de laisser l'intégralité des associations entre adresses MAC et adresses IP se réaliser dynamiquement. Si on souhaite néanmoins configurer un grand nombre d'associations statiques, il sera intéressant de renseigner un fichier **/etc/ethers**, et d'appeler la commande **arp** avec l'option **-f**.

Format du fichier /etc/ethers

```
adresse_mac1 adresse_ip1
adresse_mac2 adresse_ip2
...
adresse_macn adresse_ipn
```

Exploitation de la commande arp

On utilise ici la commande **arp** pour afficher le contenu du cache arp avant et après activité. On affecte ensuite manuellement une adresse MAC à une adresse IP, puis on prend en compte le contenu du fichier **/etc/ethers** pour configurer plusieurs associations.

```
alpha:~# arp -n
alpha:~# ping 192.168.199.1
PING 192.168.199.1 (192.168.199.1) 56(84) bytes of data.
(...)
alpha:~# arp -n
Address          HWtype  HWaddress          Flags Mask  Iface
192.168.199.1    ether   08:00:27:e4:07:62   C           eth0
alpha:~# arp -s 192.168.199.222 00:01:02:a1:b2:b3
alpha:~# arp -n
Address          HWtype  HWaddress          Flags Mask  Iface
192.168.199.222  ether   00:01:02:a1:b2:b3   CM          eth0
192.168.199.1    ether   08:00:27:e4:07:62   C           eth0
alpha:~# cat /etc/ethers
00:00:00:01:02:03 192.168.199.33
00:00:00:01:02:04 192.168.199.34
00:00:00:01:02:05 192.168.199.35
```

```
00:00:00:01:02:06 192.168.199.36
alpha:~# arp -f
alpha:~# arp -n
Address      HWtype  HWaddress      Flags Mask  Iface
192.168.199.222 ether    00:01:02:a1:b2:b3  CM          eth0
192.168.199.33 ether    00:00:00:01:02:03  CM          eth0
192.168.199.35 ether    00:00:00:01:02:05  CM          eth0
192.168.199.34 ether    00:00:00:01:02:04  CM          eth0
192.168.199.36 ether    00:00:00:01:02:06  CM          eth0
192.168.199.1 ether    08:00:27:e4:07:62  C           eth0
alpha:~#
```

b. TCP Wrappers

Il est possible de gérer les accès à un système Linux selon les adresses IP ou les noms d'hôtes des clients. On peut gérer une liste de « tous ceux qui sont autorisés », ou bien une liste de « tous ceux qui sont interdits ». Même si les techniques modernes d'intrusion et de piratage rendent ce type de contrôle d'accès presque insignifiant, cela reste tout de même une forme de contrôle rudimentaire qui peut décourager les touristes.

L'implémentation TCPWrappers utilisée sur les systèmes Linux s'appuie sur la bibliothèque libwrap.

Les deux fichiers permettant ce contrôle sont **/etc/hosts.allow** pour les clients autorisés, et **/etc/hosts.deny** pour les clients non autorisés. Ils sont lus par le démon **tcpd** qui appliquera les contrôles d'accès en conséquence. De par leur principe de fonctionnement, ces fichiers devraient être utilisés indépendamment : si on autorise certains hôtes à se connecter, cela signifie que tous les autres sont interdits, et donc le fichier d'interdiction perd de son intérêt. Si toutefois les deux fichiers étaient présents sur un système, seul le fichier **/etc/hosts.allow** serait appliqué, et le fichier **/etc/hosts.deny** serait ignoré.

Format des fichiers hosts.allow et hosts.deny

```
service: clients
```

TCP Wrappers : fichiers de contrôle d'accès	
service	Le nom du service dont l'accès est contrôlé. ALL est une valeur courante qui représente tous les services éligibles.
clients	Nom DNS ou adresse IP des clients. Plusieurs valeurs peuvent être renseignées séparées par des espaces. supporte de nombreux jokers et formats. ALL est une valeur courante qui représente toutes les adresses IP.

Exemple de fichier hosts.allow

Notez le premier exemple dont l'adresse se termine par un point. Cette syntaxe un peu particulière permet de désigner les adresses dont la partie précédant le point concorde.

```
# Toutes les adresses commençant par 10.1 sont éligibles
ftp:10.1.
# Seule l'adresse 172.12.5.28 peut se connecter
sshd: 172.12.5.28
# Toutes les adresses du réseau 192.168.1.0 peuvent se connecter
ALL: 192.168.1.0/255.255.255.0
```

4. Configuration Wi-Fi

Les distributions et les bureaux graphiques fournissent des utilitaires graphiques pour l'administration des réseaux Wi-Fi dont l'utilisation est intuitive. Nous allons donc voir ici comment configurer pas à pas une connexion Wi-Fi en lignes de commandes. Les principaux outils seront **ifconfig**, **iwconfig**, et **iwlist**.

a. Détermination de l'interface Wi-Fi

Nous savons que la commande **ifconfig -a** permet d'observer toutes les interfaces réseaux présentes sur un

système, même si elles ne sont pas activées. Une carte Wi-Fi doit donc forcément se trouver dans la liste renvoyée par cette commande. Toutefois, si le système comporte plusieurs cartes réseau, une commande spécifique nous permettra d'affiner notre choix.

Visualisation des interfaces Wi-Fi avec iwconfig

Toutes les interfaces renvoyant une référence à 802.11 sont des interfaces Wi-Fi. Dans cet exemple, c'est la carte eth1.

```
toto@ubuntu:~$ iwconfig
lo          no wireless extensions.

eth0        no wireless extensions.

eth1        IEEE 802.11  Nickname:""
            Access Point: Not-Associated
            Link Quality:5  Signal level:0  Noise level:166
            Rx invalid nwid:0  invalid crypt:0  invalid misc:0

vboxnet0    no wireless extensions.
```

b. Visualisation des réseaux disponibles

La commande **iwlist** permet de faire l'inventaire des réseaux disponibles.

Syntaxe de la commande iwlist pour la visualisation des réseaux environnants

```
iwlist interface scan
```

Où *interface* est le nom de la carte réseau Wi-Fi, et scan le paramètre qui indique la nature de l'action à opérer.

Exemple de scan avec iwlist

Dans cet exemple, on voit que deux réseaux sont disponibles. Le premier est émis par un point d'accès dont l'adresse MAC est 00:0A:66:13:E7:01, fonctionnant en 802.11g (2,4 GHz et 54 Mb/s), et dont le SSID est pifou. L'encryption est réalisée en WPA-TKIP. Le deuxième provient d'un point d'accès dont l'adresse MAC est CA:9D:2E:E6:B7:56, émettant aussi en 802.11g, avec le ssid hotspot et sans aucune sécurité.

```
toto@ubuntu:~$ sudo iwlist eth1 scan
eth1      Scan completed :
          Cell 01 - Address: 00:0A:66:13:E7:01
                    ESSID:"pifou"
                    Mode:Managed
                    Frequency=2.437 GHz (Channel 6)
                    Quality:5/5  Signal level:-50 dBm  Noise level:-78 dBm
                    IE: WPA Version 1
                        Group Cipher : TKIP
                        Pairwise Ciphers (1) : TKIP
                        Authentication Suites (1) : PSK
                    Encryption key:on
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 18 Mb/s
                               24 Mb/s; 36 Mb/s; 54 Mb/s; 6 Mb/s; 9 Mb/s
                               12 Mb/s; 48 Mb/s
          Cell 02 - Address: CA:9D:2E:E6:B7:56
                    ESSID:"hotspot"
                    Mode:Managed
                    Frequency=2.427 GHz (Channel 4)
                    Quality:1/5  Signal level:-83 dBm  Noise level:-84 dBm
                    Encryption key:off
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 9 Mb/s
                               18 Mb/s; 36 Mb/s; 54 Mb/s; 6 Mb/s; 12 Mb/s
                               24 Mb/s; 48 Mb/s
```

c. Connexion à un réseau non sécurisé

Une fois le réseau déterminé, on peut s'y connecter par la commande iwconfig.

Association à un réseau sans fil ouvert

```
iwconfig interface essid nom_ssid
```

Où *interface* représente l'interface réseau Wi-Fi gérée par le système, et *nom_ssid* le nom du réseau Wi-Fi (*Service Set Identifier*) auquel on souhaite se connecter.