

interfaces : configuration du réseau

- Objet : Le fichier : `/etc/network/interfaces`
- Niveau requis :
[débutant](#), [avisé](#)
- Commentaires : *Écrire la configuration de connexion de votre réseau.*
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi :
[à-tester](#), [à-compléter](#)
 - Création par [captfab](#) le 16/09/2013
 - Testé par ... le ...
- Commentaires sur le forum : [ici](#)¹⁾

Prérequis

 À déplacer dans une section « Configuration du réseau ».

Notion d'interface

Une interface (voir http://wiki.chezlefab.net/tuto_nix/config_network) va consister en :

- une adresse IP (via laquelle les machines pourront joindre la vôtre),
- un masque de sous-réseau (indiquant la taille du réseau),
- une adresse de diffusion (permettant de contacter toutes les machines du réseau) et
- une adresse de réseau (indiquant le point commun entre les adresses des machines).

Ces informations vont être à spécifier par l'utilisateur, ou détectées automatiquement.



On pourra aussi déclarer d'autres informations dans une interface.

Plutôt que RJ45 qui est une norme de connecteur, le réseau filaire est **ethernet** (on peut avoir la norme BNC).

Le fichier de configuration

Le fichier `/etc/network/interfaces` sert à paramétrer l'accès de votre ordinateur à un réseau. Nous pouvons le configurer manuellement, de façon simple et basique, pour que votre ordinateur se connecte à internet, que ce soit en filaire²⁾ ou en sans fil³⁾, par l'intermédiaire d'un routeur, plus communément appelé *box*.

Conflits

Lorsque nous éditons notre propre configuration réseau, les outils graphiques comme [NetworkManager](#) ou Wicd doivent être supprimés⁴⁾.

```
apt remove network-manager wicd
```

Fichier networkmanager.conf

On peut supprimer le conflit de network-manager en modifiant le fichier networkmanager.conf ainsi :
Changer [ifupdown]managed=false par =true.

Merci à **ragamatrix** sur le salon df irc. 😊

Nommage des interfaces

Lorsque nous avons plusieurs cartes réseau sur le même PC, nous pouvons aussi gérer le numéro attribué à chacune d'elle. Cela se fait au travers du fichier /etc/udev/rules.d/70-persistent-net.rules. Voir :

- [/etc/udev](#)

Rappels rapides

Les interfaces

Pour lister les interfaces réseau **disponibles**, utilisons la commande [ifconfig](#) :

```
/sbin/ifconfig -a
```

Cette commande indique chez moi :

[retour de la commande "/sbin/ifconfig -a"](#)

```
eth0      Link encap:Ethernet  HWaddr xx:xx:xx:xx:xx:xx
          inet adr:xxx.xxx.x.xx  Bcast:xxx.xxx..1.255
Masque:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:35162 errors:0 dropped:0 overruns:0 frame:0
          TX packets:36695 errors:0 dropped:0 overruns:0 carrier:1
          collisions:0 lg file transmission:1000
          RX bytes:19067074 (18.1 MiB)  TX bytes:5196716 (4.9 MiB)
          Interruption:17

eth1      Link encap:Ethernet  HWaddr xx:xx:xx:xx:xx:xx
          inet adr:xxx.xxx.x.xx  Bcast:xxx.xxx..1.255
Masque:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

```

RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 lg file transmission:1000
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
Interruption:18

eth2    Link encap:Ethernet  HWaddr xx:xx:xx:xx:xx:xx
        inet adr:xxx.xxx.x.xx  Bcast:xxx.xxx..1.255
Masque:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 lg file transmission:1000
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
Interruption:17

lo      Link encap:Boucle locale
        inet adr:127.0.0.1  Masque:255.0.0.0
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
RX packets:81 errors:0 dropped:0 overruns:0 frame:0
TX packets:81 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 lg file transmission:0
RX bytes:8560 (8.3 KiB) TX bytes:8560 (8.3 KiB)
```

- L'interface `lo` est l'interface dite de *loopback*. Elle permet au système de discuter avec lui-même et est absolument indispensable. L'unique adresse sur cette interface est `127.0.0.1`.
- Les interfaces `ethX` où `X` est un entier sont associées aux cartes réseau.
- Les interfaces `athX` ou `wlanX`, où `X` est un entier, sont associées aux cartes WiFi.
- Les interfaces `tunX` ou `tapX` sont souvent associées à des VPN.
- Les interfaces `pppX` pour les modems...
- Les interfaces `usbX` sont liées aux connexions réseau via USB, comme avec les *smartphones*.
- Les interfaces `brX` correspondent à des *bridges*, c'est à dire la fusion de plusieurs interfaces en une seule.

Pour lister les interfaces réseau **allumées**, la commande est la même sans l'option `-a` :

```
/sbin/ifconfig
```

Les routes

Pour lister les routes actives, on utilise la commande suivante :

```
/sbin/route -n
```

La route par défaut est celle menant à votre routeur et dirigeant les paquets vers internet...

Les serveurs DNS

Pour lister les résolveurs DNS actifs, on utilise la commande :

```
cat /etc/resolv.conf
```

Les lignes commençant par `nameserver` désignent les serveurs DNS. Voir : [resolv.conf](#).

Édition du fichier interfaces



RAPPEL : Comme nous sommes en phase d'administration de la machine, nous avons très fréquemment besoin des droits de super-utilisateur. Les commandes nécessitant ces droits sont inscrites dans un cadre rouge. Il est d'usage de se placer dans un shell root lorsque l'on a besoin d'effectuer ce genre de tâches.



Tous les fichiers de configuration (config) de Linux (sauf les fichiers de configuration utilisateurs) se trouvent dans le répertoire [/etc](#).



ATTENTION ! Nous ne listons plus, nous modifions maintenant...

On va modifier un fichier de configuration crucial pour le système, si on fait des bêtises on peut empêcher toute connexion réseau de la part du système, voire plus si on casse l'interface `loopback`.

Faites préalablement une copie de sauvegarde (voir la commande `cp`), genre :

```
cp /etc/network/interfaces /etc/network/interfaces-old
```

Arrêter le réseau ainsi :

```
service networking stop
```

Éditer le fichier `interfaces` avec l'éditeur [nano](#) :

```
nano /etc/network/interfaces
```

Nous obtenons (au moins) :

[interfaces](#)

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
```

```
auto lo
iface lo inet loopback
```

Après édition, démarrer le réseau ainsi :

```
service networking start
```

Configuration automatique via DHCP

Si votre interface est branchée sur un routeur, un téléphone, ou n'importe quel machin fournissant une adresse automatique via DHCP, le code à ajouter pour configurer l'interface est le suivant (mettons que l'on configure une carte réseau d'interface eth0) :

</etc/network/interfaces>

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



Pensez à laisser les directives concernant l'interface *loopback* !

1. La première ligne indique que le câble réseau peut être branché à n'importe quel moment.
2. La seconde indique qu'il faut alors lancer des requêtes DHCP pour interroger un serveur et se configurer automatiquement.

On relance alors l'interface avec les commandes :

```
ifdown eth0
```

```
ifup eth0
```

À adapter si l'interface que vous configurez n'est pas eth0.

Vérifiez la connexion :

```
/sbin/ifconfig eth0
```

Vous devez pouvoir lire, par exemple pour ma connexion :

[extrait du retour de la commande "/sbin/ifconfig eth0"](#)

```
eth0      Link encap:Ethernet  HWaddr 00:15:f2:4d:a0:d1
          inet adr:192.168.1.21  Bcast:192.168.1.255
Masque:255.255.255.0
...

```

L'attribution d'une adresse IP étant confirmée,

[extrait de l'extrait ci-dessus](#)

```
...  
    inet adr:192.168.1.21  
...
```

nous sommes donc bien connectés : le serveur DHCP a attribué l'adresse IP 192.168.1.21 à l'interface réseau eth0 de notre machine.

Configuration manuelle

Si nous ne disposons pas de serveur DHCP ou si nous voulons imposer manuellement une adresse IP, il nous est possible d'attribuer à notre interface réseau une adresse **statique** (*static*).

Configuration IP

Par exemple, en supposant toujours que nous configurons l'interface eth0, éditons le fichier /etc/network/interfaces de sorte à lui ajouter ces lignes :

[/etc/network/interfaces](#)

```
allow-hotplug eth0  
iface eth0 inet static  
    address 192.168.1.50  
    broadcast 192.168.0.255  
    netmask 255.255.255.0
```



Pensez à laisser les directives concernant l'interface *loopback* !

1. La première ligne indique que le câble réseau peut être branché à n'importe quel moment.
2. La seconde ligne indique que l'on va configurer l'interface eth0 manuellement.
3. La troisième ligne précise l'adresse que l'on veut donner à cette interface.
4. La quatrième ligne indique l'adresse de diffusion pour ce réseau.
5. La cinquième ligne indique le masque de réseau.

Les machines d'un réseau local sont souvent configurées avec pour adresse 192.168.X.Y.

En fait, pour un réseau local donné :

- l'adresse du réseau sera 192.168.X.0,
- les machines auront pour IP 192.168.X.1 à 192.168.X.254,
- l'adresse de diffusion sera 192.168.X.255,
- et le masque de réseau indique le préfixe⁵⁾ commun à tous les membres du réseaux :

255.255.255.0.

Pour le choix des IP fixes voir la rubrique IP de cet excellent site :
<http://olivieraj.free.fr/fr/linux/information/firewall/>



(merci à **fmr** 😊 de nous l'avoir indiqué...)

et aussi :

<http://www.commentcamarche.net/contents/993-nat-translation-d-adresses>

(sur la collaboration de **freddec**)

On relance alors l'interface avec les commandes :

```
ifdown eth0
```

```
ifup eth0
```

À adapter si l'interface que vous configurez n'est pas eth0.

Vous pouvez utiliser la commande [ping](#) pour vérifier la bonne marche de votre configuration.

Par exemple, si vous avez une autre machine dont l'adresse IP est 192.168.1.42 et qui est donc dans le même réseau, elle devrait répondre à un :

```
ping -c 4 192.168.1.42
```



L'option `-c 4` de la commande `ping` permet de limiter à 4 le nombre de requêtes ICMP de type *ping* à envoyer sur le réseau.

Passerelle

Pour pouvoir accéder à l'internet, il faudra indiquer à votre machine quelle est l'adresse IP du routeur qui va permettre cet accès.

Ce routeur sera donc une passerelle (*gateway*) entre le réseau local, auquel votre machine est connectée, et le réseau internet.

Dans le fichier `/etc/network/interfaces`, il faudra donc indiquer à l'interface réseau (`eth0`) de votre machine, qui est reliée à ce routeur, l'adresse IP de cette passerelle en utilisant la directive `gateway` :

[/etc/network/interfaces](#)

```
allow-hotplug eth0
iface eth0 inet static
    address 192.168.1.50
```

```
broadcast 192.168.1.255
netmask 255.255.255.0
gateway 192.168.1.1
```

À ce stade, vous devez pouvoir envoyer des paquets aux IP extérieures.
Par exemple, le serveur DNS Google, dont l'adresse IP sur le réseau internet est 8.8.8.8, devrait répondre à une requête ping :

```
ping -c 4 8.8.8.8
```

Résolution DNS

Enfin, il vous reste à apprendre à votre ordi à résoudre les noms de domaine, c'est à dire à transformer `debian.org` en `176.34.135.167`. Pour ce faire, il faut configurer le fichier de résolution DNS.

Vous pouvez le faire à la main en éditant votre fichier `resolv.conf` (voir [resolv.conf](#)). Pour configurer cela automatiquement, une fois que votre réseau fonctionne, installez le paquet `resolvconf` :

```
apt-get install resolvconf
```

Et ajoutez à votre interface `eth0` la directive `dns-nameservers` bien renseignée, ce qui nous donne l'interface suivante :

[/etc/network/interfaces](#)

```
allow-hotplug eth0
iface eth0 inet static
    address 192.168.1.50
    broadcast 192.168.1.255
    netmask 255.255.255.0
    gateway 192.168.1.1
    dns-nameservers 208.67.222.222
```

Dans laquelle vous aurez remplacé `208.67.222.222` par votre serveur DNS préféré.

Cette fois-ci, vous pouvez tester que le tout fonctionne en pingant un nom de domaine, comme :

```
ping -c 4 example.net
```

Plusieurs cartes réseau

Nous pouvons aussi utiliser simultanément plusieurs cartes réseau dans le même bouzin.

Si vous avez **deux cartes réseau** installées sur la même bécane, il suffit de brancher les

connecteurs, un câble sur le réseau local et un câble sur le réseau internet.
Ensuite vous en mettez une en dhcp pour la connexion internet et la seconde en static sur un réseau local.

Pour en mettre une (eth0 par exemple) sur internet et l'autre (eth1...) sur le réseau local, nous configurons interfaces ainsi :

Éditons le fichier /etc/network/interfaces, et l'on rédige :

[/etc/network/interfaces](#)

```
# The primary network interface
allow-hotplug eth0
iface eth0 inet dhcp

# The secondary network interface
allow-hotplug eth1
iface eth1 inet static
address 192.168.1.50
broadcast 192.168.255.255
netmask 255.255.255.0
```

Bien entendu, l'IP (192.168.1.50) indiquée ici est arbitraire, à vous de le déterminer comme vous l'entendez, voir plus bas pour les IP.

Le cas du WiFi

Pour configurer le WiFi, des options supplémentaires sont nécessaires, voir :

- [interfaces : configuration WiFi](#)

Les options de démarrage

Regardons de plus près la ligne `auto-hotplug eth0`

Pas de démarrage auto

Si tu ne mets aucune ligne avant :

```
iface...
```

rien ne se passe au *boot*, l'interface ne démarre qu'avec :

```
ifup eth0
```

Lancement au démarrage

Si tu mets `auto eth0`, l'interface monte au démarrage de la machine ou au redémarrage du réseau.

Si la carte est absente, le système finit par t'injurier :

```
failed to bring up interface eth0
```

mais continue son *boot*.

Ainsi, normalement avec `auto eth0`, ton interface (la connexion) doit monter au démarrage (*boot*).

L'insertion de la carte ou de la fiche RJ45 à chaud ne produit rien, sauf si tu redémarres l'interface avec :

```
service networking restart
```

Lancement au branchement

`allow-hotplug` à la place de `auto` permet de démarrer automatiquement l'interface lorsque tu insères le câble.

Remarque

`allow-hotplug` devient le comportement par défaut de Debian.

Avant, on faisait tout au *boot* (démarrage) le `xorg.conf`, les interfaces, les périphériques...

Maintenant la faveur est tout en *hotplug*, branchement à chaud sans manipulation de la configuration de base.



Si vous commentez la ligne `auto` ou celle `allow`, il n'y aura alors pas de connexion active au démarrage, cependant, il suffira d'un simple `ifup` pour que la connexion s'établisse.

Suggéré par **raleur** sur le forum, là :

- <https://debian-facile.org/viewtopic.php?pid=156710#p156710> 😊

Liens et remerciements

Il reste que nous avons là :

- http://wiki.chezlefab.net/tuto_nix/config_network

Un super bon tuto pour aller bien plus loin que cette légère introduction à la configuration de son réseau !

Hourra pour le *captnfab* !

Et comme le fait judicieusement remarquer **cthuluh**, il y a la commande :

```
man interfaces
```

En anglais bien sûr ! 😊

Liens utiles

Merci à **Haricophile** et à **Geko** pour ces liens :

- <https://lists.debian.org/debian-user-french/2005/10/msg01775.html> (in french... cool !)
- <https://lists.debian.org/debian-doc/2008/02/msg00010.html> (en anglais lisible...)

Liens à suivre

- [Sécuriser sa Debian](#)

Astuces

Attribution d'un DNS fixe via DHCP

 À déplacer dans la page de dhclient une fois créée.

Une façon moins orthodoxe de définir un serveur de nom de domaine pour l'ensemble du système est de le paramétrer directement dans la configuration de dhclient. Cela évite que les infos de serveur de nom de domaine qu'enverra le routeur lors de la négociation DHCP ne viennent modifier le serveur de nom que vous avez défini.

Pour ce faire, il convient d'éditer le fichier `etc/dhcp/dhclient.conf` et d'ajouter la ligne suivante avant la liste des informations devant être demandées au routeur (ligne commençant par `request`) :

```
supersede domain-name-servers 208.67.220.220;
```

(L'IP renseignée pour l'exemple est celle des serveurs d'OpenDNS)

Une fois votre système redémarré, le fichier `etc/resolv.conf` contiendra l'IP de DNS définie.

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

2)

protocole : **ethernet** *RJ45, BNC, ...*

3)

protocole : **wifi**

4)

Voir : [les outils apt ici](#) et [les éditeurs de fichier là](#)

5)

bit-à-bit

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/doc:reseau:interfaces>



Last update: **03/06/2023 08:16**