

# Réseau

## Netconfig

En eux-mêmes les ordinateurs ne sont pas très intéressants. Bien sûr, vous pouvez y installer quelques jeux, mais cela ne fait que les changer en consoles. Aujourd'hui les ordinateurs ont besoin de pouvoir parler entre eux, ils ont besoin d'être en réseau. Que ce soit pour mettre en place un réseau professionnel avec des centaines ou des milliers d'ordinateurs ou juste établir une connexion à Internet avec un simple PC, Slackware rend cela simple et facile. Ce chapitre devrait vous apprendre comment mettre en place les réseaux les plus courants. Pour ce qui concerne les réseaux sans fil (*wireless*) cela sera discuté dans le chapitre suivant, même si beaucoup de ce que vous pourrez lire ici s'y appliquera également.

Il existe bon nombre de façons différentes de connecter votre ordinateur à un réseau ou à Internet mais cela se divise en deux catégories : statique ou dynamique. Les adresses statiques sont consistantes, elles sont définies de manière à ne pas être changées, ou du moins pas dans l'immédiat. Les adresses dynamiques sont mouvantes, le postulat est que l'adresse sera changée dans un futur proche. D'une manière générale, tout ce qui ressemble à un serveur doit utiliser une adresse statique simplement pour que les autres machines sachent comment le contacter lorsqu'elles ont besoin de ses services. Les adresses dynamiques sont plutôt utilisées pour les postes de travail, les clients sur Internet et toute autre machine qui ne requière pas d'adresse statique pour une raison quelconque. Les adresses dynamiques sont plus flexibles mais apportent leurs propres complications.

Vous pourrez être amenés à rencontrer beaucoup de protocoles réseau, mais la plupart des gens n'ont besoin que de savoir gérer Internet Protocol (IP). Pour cette raison, nous nous concentrerons exclusivement sur IP dans ce livre.

## Configuration manuelle

Très bien, vous avez installé Slackware, vous avez configuré votre ordinateur de bureau, mais vous ne pouvez pas le connecter à Internet ou au LAN (*Local Area Network* - réseau local) de votre entreprise ; que faire ? Heureusement la solution à cette question est simple. Slackware comporte plusieurs outils pour configurer votre connexion réseau. Le premier que nous étudierons est le très puissant **ifconfig**(8), qui est utilisé pour configurer ou modifier la configuration du matériel le plus courant pour se connecter à un réseau : la carte réseau (NIC - *Network Interface Card* ou *carte Ethernet*). **ifconfig** est un outil incroyablement puissant capable de faire bien plus que d'attribuer des adresses IP. Pour une présentation complète, vous devriez lire sa page de manuel. Dans l'immédiat, nous allons juste l'utiliser pour afficher et modifier les adresses réseaux de certains contrôleurs Ethernet.

```
darkstar:~# ifconfig
lo          Link encap:Local Loopback
            inet addr:127.0.0.1  Mask:255.0.0.0
            inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING  MTU:16436  Metric:1
            RX packets:699 errors:0 dropped:0 overruns:0 frame:0
            TX packets:699 errors:0 dropped:0 overruns:0 carrier:0
```

```

collisions:0 txqueuelen:0
RX bytes:39518 (38.5 KiB) TX bytes:39518 (38.5 KiB)

wlan0    Link encap:Ethernet HWaddr 00:1c:b3:ba:ad:4c
inet addr:192.168.1.198 Bcast:192.168.1.255 Mask:255.255.255.0
inet6 addr: fe80::21c:b3ff:feba:ad4c/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:1630677 errors:0 dropped:0 overruns:0 frame:0
TX packets:1183224 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:1627370207 (1.5 GiB) TX bytes:163308463 (155.7 MiB)

wmaster0 Link encap:UNSPEC HWaddr 00-1C-B3-BA-
AD-4C-00-00-00-00-00-00-00-00
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 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

```

Comme vous pouvez clairement le voir ici, lorsqu'elle est lancée sans option, la commande **ifconfig** affichera toutes les informations sur les cartes réseau Ethernet (et les cartes sans fil Ethernet) présentes sur votre système. Ce qui est affiché ci-dessus est une connexion sans fil classique depuis mon ordinateur portable, ne soyez donc pas inquiets si cela ne correspond pas à ce que vous pouvez voir sur votre système. Toutefois, si vous ne voyez aucune interfaces ethX ou wlanX, l'interface peut être désactivée. Pour afficher toutes les cartes réseau installées, qu'elles soient actives ("*up*") ou non ("*down*"), utilisez simplement l'option *-a*.

```

darkstar:~# ifconfig -a
eth0    Link encap:Ethernet HWaddr 00:19:e3:45:90:44
UP BROADCAST MULTICAST MTU:1500 Metric:1
RX packets:122780 errors:0 dropped:0 overruns:0 frame:0
TX packets:124347 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:60495452 (57.6 MiB) TX bytes:17185220 (16.3 MiB)
Interrupt:16

lo      Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:699 errors:0 dropped:0 overruns:0 frame:0
TX packets:699 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:39518 (38.5 KiB) TX bytes:39518 (38.5 KiB)

wlan0    Link encap:Ethernet HWaddr 00:1c:b3:ba:ad:4c
inet addr:192.168.1.198 Bcast:192.168.1.255 Mask:255.255.255.0
inet6 addr: fe80::21c:b3ff:feba:ad4c/4 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:1630677 errors:0 dropped:0 overruns:0 frame:0

```

```
TX packets:1183224 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:1627370207 (1.5 GiB) TX bytes:163308463 (155.7 MiB)
```

```
wmaster0 Link encap:UNSPEC HWaddr 00-1C-B3-BA-
AD-4C-00-00-00-00-00-00-00-00-00-00
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 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

Remarquez que l'interface eth0 est maintenant listée dans le résultat. **ifconfig** peut changer les paramètres actifs d'une carte réseau. Généralement, vous souhaitez modifier l'adresse IP et le masque de sous-réseau, mais vous pouvez virtuellement modifier n'importe quel paramètre.

```
darkstar:~# ifconfig eth0 192.168.1.1 netmask 255.255.255.0
darkstar:~# ifconfig eth0
eth0      Link encap:Ethernet HWaddr 00:19:e3:45:90:44
          inet addr:192.168.1.1 Bcast:192.168.1.255 Mask:255.255.255.0
          UP BROADCAST MULTICAST MTU:1500 Metric:1
          RX packets:122780 errors:0 dropped:0 overruns:0 frame:0
          TX packets:124347 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:60495452 (57.6 MiB) TX bytes:17185220 (16.3 MiB)
          Interrupt:16
```

Si vous regardez attentivement, vous remarquerez que l'interface a maintenant l'adresse IP 192.168.1.1 et le masque de sous-réseau 255.255.255.0. Nous avons maintenant défini les paramètres essentiels pour se connecter à notre réseau, mais nous avons toujours besoin de définir une passerelle (*gateway*) par défaut et nos serveurs DNS. Pour cela, nous allons avoir besoin de quelques outils supplémentaires.

Notre prochain arrêt au pays du réseau est le tout aussi puissant **route**(8). Cet outil est responsable des modifications de la table de routage du noyau Linux, qui concerne toutes les transmissions de données vers un réseau. Les tables de routage peuvent être extrêmement complexes ou être simple et directes. La plupart des utilisateurs n'auront uniquement besoin que de définir une passerelle par défaut, c'est donc ce que nous allons vous montrer ici. Si pour quelque raison vous avez besoin d'une table de routage plus compliquée, vous êtes fortement incités à lire la page de manuel complète de **route** ainsi que d'autres références. Dans l'immédiat, jetons un œil sur la table de routage juste après avoir paramétré eth0.

```
darkstar:~# route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use
Iface
192.168.1.0 * 255.255.255.0 U 0 0 0 eth0
loopback * 255.0.0.0 U 0 0 0 lo
```

Je n'expliquerai pas tout ici, mais le principe général devrait être facile à comprendre si vous êtes un peu à l'aise avec les notions de réseau. Les champs *Destination* et *Genmask* indiquent des critères

basés sur des étendues d'adresses IP. Si une passerelle (*Gateway*) est définie, l'information, sous forme de paquets, sera envoyée vers cet hôte pour être réexpédiée. Nous pouvons aussi indiquer une interface dans le dernier champ, que l'information devra utiliser. Dans l'immédiat, nous pouvons uniquement communiquer avec les ordinateurs ayant des adresses comprises entre 192.168.1.0 et 192.168.1.255 et nous-même au travers de l'interface de bouclage (*loopback*), qui est une sorte de carte réseau virtuelle utilisée pour router l'information depuis l'ordinateur vers lui-même. Afin de pouvoir contacter le reste du monde, nous devons définir une passerelle par défaut (*default gateway*).

```
darkstar:~# route add default gw 192.168.1.254
darkstar:~# route
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use
Iface
192.168.1.0      *                255.255.255.0   U        0      0      0 eth0
loopback         *                255.0.0.0       U        0      0      0 lo
default          192.168.1.254   0.0.0.0         UG       0      0      0 eth0
```

Vous devriez immédiatement remarquer l'ajout de la route par défaut. Ceci précise quel routeur doit être utilisé pour atteindre n'importe quelle adresse qui n'est pas précisée par ailleurs dans notre table de routage. Maintenant, lorsque nous essayons de contacter disons, 64.57.102.34, l'information sera envoyée vers 192.168.1.254 qui sera responsable pour nous d'envoyer les données.

Malheureusement, cela n'est pas encore suffisant. Nous avons besoin de pouvoir convertir des noms de domaine tel que `slackware.com` en adresses IP que l'ordinateur pourra utiliser. Pour cela, nous devons utiliser un serveur DNS.

Heureusement, il est très facile de paramétrer votre ordinateur pour utiliser un serveur DNS externe (ou même interne). Vous aurez besoin d'utiliser votre éditeur de texte favori et ouvrir le fichier `/etc/resolv.conf`. Ne me demandez pas ce qu'il est advenu du **e**. Sur mon ordinateur `resolv.conf` ressemble à ceci :

```
# /etc/resolv.conf
search lizella.net
nameserver 192.168.1.254
```

La plupart des utilisateurs n'auront pas besoin de la ligne `search`. C'est utilisé pour faire correspondre les noms d'hôtes avec un nom de domaine par défaut. De cette manière, si j'essaye de me connecter à `"barnowl"`, l'ordinateur sait que je recherche `"barnowl.lizella.net"` grâce à la ligne `search`. Nous sommes plus intéressés par la ligne `nameserver`. Ceci indique à Slackware à quel serveur de nom (*domain name server* - DNS) se connecter. D'une manière générale, vous devriez toujours indiquer une adresse IP. Si vous savez quels serveurs DNS vous devez utiliser, vous pouvez juste les ajouter, un par ligne. De fait, je ne sais pas s'il existe une limite au nombre de serveurs de noms que vous pouvez indiquer dans `resolv.conf`, vous pouvez donc en ajouter autant que vous voulez <sup>1)</sup>. Une fois ceci effectué, vous devriez être capable de communiquer avec d'autres hôtes par leur nom de domaine pleinement qualifié (*fully qualified domain name* - FQDN).

Mais Alan, c'est une tonne de travail ! Je ne veux pas avoir à faire ça pour des dizaines ou des centaines de machines. Vous avez tout à fait raison et c'est pourquoi des gens plus intelligents que vous et moi ont créé DHCP. DHCP signifie *Dynamic Host Control Protocol* (protocole de configuration dynamique d'hôtes) et est un moyen pour configurer automatiquement des ordinateurs avec chacun leur adresse IP, masque de sous-réseau, passerelle et serveurs DNS. La plupart du temps, vous voudrez juste utiliser DHCP. La majorité des routeurs sans-fil, modems ADSL ou modem câble, même

des pare-feu tous disposent de serveurs DHCP qui peuvent rendre la vie plus facile. Slackware propose deux outils principaux pour se connecter à un serveur DHCP existant et peut aussi agir comme serveur DHCP pour d'autres ordinateurs. Pour l'instant, nous n'allons parler que de ce qui concerne les clients DHCP.

Le premier sur notre liste est **dhcpcd**(8), fourni avec les utilitaires d'ISC DHCP. En partant du principe que votre ordinateur est physiquement relié à votre réseau, et que vous avez un serveur DHCP opérationnel sur votre réseau, vous pouvez configurer votre carte réseau d'un coup :

```
darkstar:~/ dhcpcd eth0
```

Si tout se déroule sans accroc, votre carte réseau devrait être correctement configurée et vous devriez être capable de communiquer avec les autres ordinateurs de votre réseau, et avec Internet au sens large. Si pour une raison quelconque **dhcpcd** échoue, vous pouvez essayer **dhclient**(8).

**dhclient** est une alternative à **dhcpcd** et fonctionne de la même façon.

```
darkstar:~/ dhclient eth0
Listening on LPF/eth0/00:1c:b3:ba:ad:4c
Sending on   LPF/eth0/00:1c:b3:ba:ad:4c
Sending on   Socket/fallback
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPACK from 192.168.1.254
bound to 192.168.1.198 -- renewal in 8547 seconds.
```

Pourquoi Slackware propose deux clients DHCP ? Parfois un serveur DHCP particulier peut être défectueux et ne pas répondre correctement soit à **dhcpcd**, soit à **dhclient**. Dans les deux cas, vous pouvez toujours vous rabattre sur le second client DHCP dans l'espoir d'obtenir une réponse valide de la part du serveur. Traditionnellement, Slackware utilise **dhcpcd** et cela fonctionne dans la grande majorité des cas, mais il peut être nécessaire dans certains cas d'utiliser **dhclient** à la place. Tous deux sont d'excellents clients DHCP, utilisez donc celui qui vous convient le mieux.

## Configuration automatique avec rc.inet1.conf

Savoir configurer manuellement des interfaces est une connaissance importante, mais cela peut devenir fastidieux. Personne ne souhaite configurer manuellement sa connexion Internet à chaque démarrage du système. Plus important, vous pouvez ne pas avoir d'accès physique à votre machine lorsqu'elle démarre. Slackware simplifie cela en configurant automatiquement les cartes Ethernet (et sans-fil) au démarrage avec `/etc/rc.d/rc.inet1.conf`. Dans l'immédiat, nous nous concentrerons sur le classique réseau câblé, nous parlerons des options sans fil dans le prochain chapitre.

`rc.inet1.conf` est un fichier de configuration incroyablement puissant, capable de configurer la plupart des cartes réseau automatiquement lorsque Slackware démarre. Le fichier est truffé de commentaires utiles, mais il existe également une page de manuel qui présente son usage de manière plus approfondie. Pour commencer, nous allons regarder plusieurs options utilisées sur une de mes machines personnelles.

```
# Config information for eth0:
```

```
IPADDR[0]="192.168.1.250"  
NETMASK[0]="255.255.255.0"  
USE_DHCP[0]=""  
DHCP_HOSTNAME[0]=""  
# Plusieurs lignes sont omises.  
GATEWAY="192.168.1.254"
```

Ceci représente l'essentiel des informations nécessaires pour configurer une adresse IP fixe pour un simple contrôleur Ethernet. **netconfig** remplira habituellement ces valeurs pour une carte réseau unique pour vous. Si vous avez plusieurs cartes réseau et que vous avez besoin de toutes les activer au démarrage, vous aurez besoin d'éditer ce fichier pour ajouter des entrées de manière similaire à celle présentée ici. Pour commencer, laissez moi débiter par des rappels élémentaires.

Comme vous l'avez probablement deviné, **IPADDR[n]** est l'adresse *Internet Protocol* pour la carte réseau **n**. En général, **n** correspond à *eth0*, *eth1* et ainsi de suite, mais ce n'est pas toujours le cas. Vous pouvez indiquer ces valeurs pour correspondre à un contrôleur réseau différent avec la variable **IFNAME[n]**, mais nous garderons cela pour les [réseaux sans fil](#), car cela concerne plus les contrôleurs réseau sans fil.

De la même manière, **NETMASK[n]** est le masque de sous-réseau à utiliser par le contrôleur réseau. Si ces lignes sont laissées vides, alors aucune adresse IP fixe ne sera attribuée automatiquement à cette carte réseau. La variable **USE\_DHCP[n]** dira (bien sûr) à Slackware d'utiliser DHCP pour configurer cette interface. **DHCP\_HOSTNAME[n]** est rarement utilisé, mais certains serveurs DHCP peuvent le demander. Dans ce cas, vous devez indiquer un nom d'hôte valide. Finalement nous arrivons à la variable GATEWAY. Elle apparaît en fait bien plus bas dans le fichier que dans mon exemple, et elle gère la passerelle par défaut à utiliser. La raison à cela est liée à comment *Internet Protocol* fonctionne. Je n'irai pas dans des explications détaillées à ce sujet, il suffit de savoir qu'il ne peut y avoir qu'une route par défaut définie pour un ordinateur, quelque soit son nombre d'interfaces réseau.

Si vous avez besoin d'utiliser un adressage IP fixe, vous devez obtenir une adresse IP unique et un masque de sous-réseau pour cette interface, ainsi qu'une passerelle par défaut, et les entrer dans ce fichier. Il n'y a pas de place pour la configuration DNS dans `rc.inet1.conf`, les serveurs DNS devront donc être entrés manuellement dans `resolv.conf` comme nous l'avons vu dans la [configuration manuelle](#). Bien sûr, si vous utilisez **netconfig**, cela sera géré pour vous par ce programme. Maintenant, regardons une autre interface de mon ordinateur.

```
# Config information for eth1:  
IPADDR[1]=""  
NETMASK[1]=""  
USE_DHCP[1]="yes"  
DHCP_HOSTNAME[1]=""
```

J'indique ici à Slackware que je configure *eth1* en utilisant DHCP. Il n'y a pas besoin de configurer les variables **IPADDR[1]** ou **NETMASK[1]** lorsque vous utilisez DHCP (en fait, si elles sont définies, elles seront ignorées). Slackware se fera une joie de contacter un serveur DHCP dès que la machine aura démarré.

# Navigation

Chapitre précédent : [Emacs](#)

Chapitre suivant : [Réseau sans fil](#)

## Sources

- Source originale : <http://www.slackbook.org/beta>
- Publication initiale d'Alan Hicks, Chris Lumens, David Cantrell, Logan Johnson
- Traduction initiale de [escaflown](#)
- Traduction de [Ellendhel](#)

[slackbook](#), [networking](#), [netconfig](#), [dhcpcd](#)  
<sup>1)</sup>

la limite est définie dans `/usr/include/resolv.h` par la constante `MAXNS`, qui est de trois par défaut. NdT.

From:

<https://docs.slackware.com/> - **SlackDocs**

Permanent link:

<https://docs.slackware.com/fr:slackbook:network>

Last update: **2016/07/18 21:26 (UTC)**

