Configuration d'un réseau TCP/IP

Les systèmes Debian squeeze peuvent gérer la connexion au réseau à l'aide de démons tels que NetworkManager (NM) (paquet network-manager et les paquets associés) ou Wicd (paquet wicd et les paquets associés).

- Il sont fournis avec leur propre interface utilisateur graphique (GUI) et en ligne de commandes.
- Ils ont leur propre démon en tant que système dorsal.
- Ils permettent une connexion facile de votre système à Internet.
- Ils permettent une gestion facile de la configuration du réseau filaire ou sans fil.
- Ils nous permettent de configurer le réseau indépendamment de l'ancien paquet « `ifupdown` »

> **Note**
>
> Ne pas utiliser ces outils de configuration automatique du réseau sur un serveur. Ils ont été prévus principalement pour les utilisateurs de système de bureau tournant sur des ordinateurs portables.

http://www.debian.org/doc/manuals/debian-reference/ch05.fr.html#_the_modern_network_configuration_for_desktop

Liste des entrées de « `/etc/network/interfaces` »

entrée	**signification**
« `auto <nom_interface>` »	démarrer l'interface <nom_interface> lors du démarrage du système
« `allow-auto <nom_interface>` »	, ,
« `allow-hotplug <nom_interface>` »	démarrer l'interface <nom_interface> lorsque le noyau détecte un événement « à chaud » depuis cette interface
Les lignes qui commencent par « `iface <nom_config> …` »	définissent la configuration de réseau <nom_config>
Les lignes qui commencent par « `mapping <nom_interface_glob>` »	définissent une valeur de correspondance de <nom_config> pour l'interface correspondant à <nom_interface>
Une ligne commençant par le signe « **#** »	est traitée comme commentaire et ignorée (les commentaire de fin de ligne ne sont **pas** pris en charge)
Une barre oblique inversée (« \ », « back slash » en anglais) en fin de ligne	étend la configuration à la ligne suivante

Une interface réseau avec une adresse IP fixe est configurée en créant de la manière suivante une entrée de configuration du fichier « `/etc/network/interfaces` » :

```
allow-hotplug eth0
iface eth0 inet static
 address 192.168.11.100
 netmask 255.255.255.0
 gateway 192.168.11.1
```

```
 dns-domain example.com
 dns-nameservers 192.168.11.1
```

Lorsque le noyau de Linux détecte l'interface physique `eth0`, l'entrée **`allow-hotplug`** permet à `ifup` d'activer l'interface en utilisant l'IP statique pour la configurer.

Connexion WLAN wifi

Liste d'acronymes pour le WLAN

acronyme	en entier	signification
NWID	ID du réseau	ID du réseau sur 16 bits utilisées par les réseaux WaveLAN pre-802.11 (complètement dépassé)
(E)SSID	Service Set Identifier (étendu)	nom de réseau des Points d'accès sans fil (AP) interconnectés pour former un réseau local sans fil 802.11, identifiant de domaine
WEP, (WEP2)	Wired Equivalent Privacy (confidentialité équivalente à un réseau câblé)	première génération de norme de chiffrement sans fil sur 64 bits (128 bits) avec une clé sur 40 bits (dépassé)
WPA	Wi-Fi Protected Access (Accès Wi-Fi protégé)	seconde génération de norme de chiffrement sans fil (la plus grande partie de 802.11i), compatible avec WEP
WPA2	Wi-Fi Protected Access 2 (Accès protégé Wi-FI 2)	troisième génération de norme de chiffrement sans fil (802.11i complète), non compatible avec WEP

réseau local sans fil avec WPA/WPA2

Vous devrez installer le paquet `wpasupplicant` afin de prendre en compte le WLAN avec les nouveaux protocoles WPA/WPA2.

Dans le cas d'IP fournies par DHCP sur une connexion WLAN, l'entrée du fichier « `/etc/network/interfaces` » doit être similaire à la suivante :

```
allow-hotplug ath0
iface ath0 inet dhcp
 wpa-ssid zonemaison
 # la clé psk hexadécimale est encodée depuis une phrase de passe en texte clair
 wpa-psk 000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f
```

réseau local sans fil avec WEP

Vous devrez installer le paquet `wireless-tools` pour prendre en charge le WLAN avec l'ancien protocole WEP. (Votre routeur grand public peut encore utiliser cette infrastructure non sûre mais c'est mieux que rien).

Dans le cas d'IP fournies par DHCP sur une connexion WLAN, l'entrée du fichier « `/etc/network/interfaces` » doit être similaire à la suivante :

```
allow-hotplug eth0
iface eth0 inet dhcp
 wireless-essid Maison
 wireless-key1 0123-4567-89ab-cdef
```

```
wireless-key2 12345678
wireless-key3 s:mot_de_passe
wireless-defaultkey 2
wireless-keymode open
```

Consultez « `/usr/share/doc/wireless-tools/README.Debian` ».

Les adresses IP privées

Voici **les adresses privées** que vous pouvez prendre en fonction de votre réponse.

1. Classe C : 192.168.0.0 - 192.168.255.0 : 255.255.255.0
2. Classe B : 172.16.0.0 - 172.31.0.0 : 255.255.0.0
3. Classe A : 10.0.0.0 : 255.0.0.0

Assignation du nom de machine

Pour beaucoup d'application réseau, il est souvent important de définir un nom de machine.

Pour cela vous pouvez utiliser la commande hostname :

```
hostname VotreMachine
```

Par exemple

```
hostname Zorro
(où "Zorro" est le nom de ma machine)

On retrouve ce nom dans le fichier /etc/hostname
```

Résolution du nom d'hôte

La résolution du nom d'hôte est actuellement prise en charge aussi par le mécanisme NSS (Name Service Switch). Le flux de cette résolution est le suivant :

1. Le fichier « `/etc/nsswitch.conf` » avec une entrée comme « `hosts: files dns` » donne l'ordre de la résolution du nom d'hôte.
2. La méthode `files` est d'abord appelée. Si le nom d'hôte est trouvé dans le fichier « `/etc/hosts` », elle retourne toutes les adresses valables qui y correspondent et quitte.
3. La méthode `dns` est appelée. Si le nom d'hôte est trouvé par une requête au Système de noms de domaine Internet (DNS) (« Internet Domain Name System ») identifié par le fichier « `/etc/resolv.conf` », elle retourne toutes les adresses valables correspondantes et quitte.

Par exemple, « `/etc/hosts` » ressemble à ce qui suit :

```
127.0.0.1 localhost
127.0.1.1 <nom_hote>.<nom_domaine> <nom_hote>

# Les lignes suivantes servent pour les machines pouvant utiliser 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
```

Chaque ligne commence par une adresse IP et est suivie du nom d'hôte associé.

Nom de l’interface réseau

Le nom de l’interface réseau, par exemple `eth0`, est assigné dans le noyau Linux à chaque matériel lorsqu'il est trouvé.

Configuration

Les systèmes Debian `squeeze` peuvent gérer la connexion au réseau à l’aide de démons tels que **NetworkManager** (NM) (paquet network-manager et les paquets associés) ou **Wicd** (paquet wicd et les paquets associés).

> Note
>
> Ne pas utiliser ces outils de configuration automatique du réseau sur un serveur. Ils ont été prévus principalement pour les utilisateurs de système de bureau tournant sur des ordinateurs portables.

Essentiellement, la configuration réseau pour l’ordinateur de bureau est faite de la manière suivante :

1. Rendez l’utilisateur du bureau, par exemple `toto`, membre du groupe « `netdev` »
2. Gardez la configuration de « `/etc/network/interfaces` » aussi simple que possible comme ce qui suit :

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

3. Redémarrez NM ou Wicd de la manière suivante :

```
$ sudo /etc/init.d/network-manager restart

$ sudo /etc/init.d/wicd restart
```

4. Configurez votre réseau à l’aide d'une interface graphique.

La connexion DHCP avec Ethernet

Le réseau local est habituellement configuré par un serveur de protocole de configuration dynamique de l’hôte (DHCP) (« **dynamic host configuration protocol** »).

les PC sur le réseau local sont connectés à internet par l’intermédiaire d'une passerelle par traduction d'adresse réseau (NAT)Dans ce cas, les interfaces réseau des PC sur le réseau local sont servies avec des adresses IP statiques ou par un serveur DHCP.

Vous pouvez (ré)initialiser l’interface réseau simplement par « **`sudo ifdown eth0;sudo ifup eth0`** ».

Programmes de bas niveaux

Il existe deux types de programmes de bas niveau pour la configuration du réseau sous un système Linux

- Les programmes net-tools anciens (ifconfig(8), …) provienne du système de réseau NET-3 de Linux. La plupart d'entre-eux sont aujourd'hui obsolètes.
- Les nouveau programmes Linux iproute2 (ip(8), …) représentent le système actuel de gestion du réseau sous Linux.

`iproute`

Table de conversion depuis les commandes obsolètes `net-tools` vers les nouvelles commandes `iproute2`

http://www.debian.org/doc/manuals/debian-reference/ch05.fr.html#_iproute2_commands

net-tools obsolètes	**nouveau iproute2, etc.**	**manipulation**
ifconfig(8)	`ip addr`	adresse de protocole (IP ou IPv6) d'un périphérique
route(8)	`ip route`	entrée de la table de routage
arp(8)	`ip neigh`	entrée de cache ARP ou NDISC
`ipmaddr`	`ip maddr`	adresse multicast
`iptunnel`	`ip tunnel`	tunnel sur IP
nameif(8)	ifrename(8)	nommer les interfaces réseau en se basant sur l'adresse MAC
mii-tool(8)	ethtool(8)	paramétrage du périphérique Ethernet

voir :http://www.inetdoc.net/guides/lartc/index.html

et aussi http://irp.nain-t.net/doku.php/start

Les noyaux Linux des séries 2.2 et plus ont un sous-système réseau complètement réécrit.
Linux possède un système d'allocation de bande passante appelé Contrôle de trafic (*Traffic Control*). Ce système permet de classer, ranger par ordre de priorité, partager et limiter le trafic entrant et sortant.
Le paquet concerné s'appelle **iproute** et a normalement été installé.
L'outil **ip** est central, et nous allons lui demander de nous montrer les interfaces.

les liens – interfaces `ip link list`

```
ip link list
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
qlen 1000
    link/ether 00:1c:c0:f8:5f:5f brd ff:ff:ff:ff:ff:ff
3: pan0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN
    link/ether 62:87:ec:b7:0e:b4 brd ff:ff:ff:ff:ff:ff
```

La première interface que nous voyons est l'interface `loopback`. La taille de MTU (unité maximum de transmission) est de 16436 octets.
La deuxième, la vrai, est **eth0**
Notons l'absence d'adresses IP. Iproute déconnecte les concepts de « liens » et « d'adresses IP ».

Nos adresses IP `ip address show/list`

```
ip address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
qlen 1000
    link/ether 00:1c:c0:f8:5f:5f brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.254/24 brd 192.168.1.255 scope global eth0
    inet6 fe80::21c:c0ff:fef8:5f5f/64 scope link
       valid_lft forever preferred_lft forever
3: pan0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN
    link/ether 62:87:ec:b7:0e:b4 brd ff:ff:ff:ff:ff:ff
```

Examinons l'interface `eth0` de plus près. Il est dit qu'elle est reliée à l'adresse internet `192.168.1.254/24`. Qu'est-ce que cela signifie ? Le /24 désigne le nombre de bits réservés à l'adresse réseau. Il y a 32 bits, donc il y a 24 bits pour désigner une partie de notre réseau `192.168.1.0`. Les 8 derniers bits repèrent des machines directement connectées à cette interface. Le masque est 255.255.255.0.

Donc,par xemple, `192.168.1.2` est directement disponible sur `eth0.`

Se limiter à eth0

```
ip addr show eth0
```

Les routes `ip route show/list`

```
ip route show
192.168.1.0/24 dev eth0  proto kernel  scope link  src
192.168.1.254  metric 1
default via 192.168.1.1 dev eth0  proto static
```

`192.168.1.1` notre passerelle par défaut

ARP `ip neigh show/list`

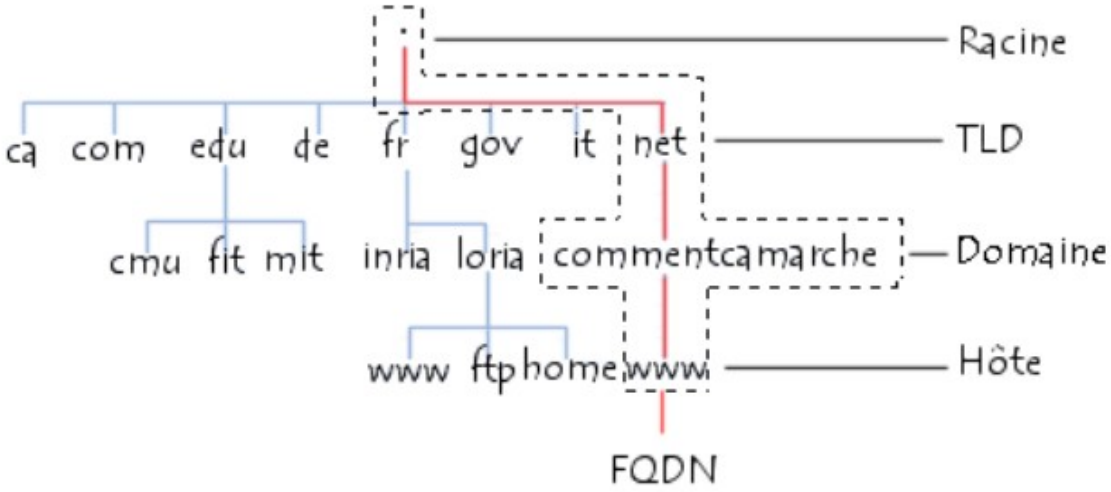
ARP est le Protocole de Résolution d'Adresse (*Address Resolution Protocol*). Il est décrit dans le RFC 826. ARP est utilisé par une machine d'un réseau local pour retrouver l'adresse matérielle (la localisation) d'une autre machine sur le même réseau. Les machines sur Internet sont généralement connues par leur nom auquel correspond une adresse IP. C'est ainsi qu'une machine sur le réseau `foo.com` est capable de communiquer avec une autre machine qui est sur le réseau `bar.net`. Une adresse IP, cependant, ne peut pas vous indiquer la localisation physique de la machine. C'est ici que le protocole ARP entre en jeu.

```
ip neigh show
192.168.1.1 dev eth0 lladdr e8:be:81:04:e5:60 REACHABLE
```

DNS

http://www.commentcamarche.net/contents/internet/dns.php3

La structuration du système DNS s'appuie sur une structure arborescente dans laquelle sont définis des domaines de niveau supérieurs (appelés **TLD**, pour *Top Level Domains*), rattachés à un noeud racine représenté par un point.



On appelle « **nom de domaine** » chaque noeud de l'arbre. Chaque noeud possède une étiquette (en anglais « *label* ») d'une longueur maximale de 63 caractères.

L'ensemble des noms de domaine constitue ainsi un arbre inversé où chaque noeud est séparé du suivant par un point (« . »).

L'extrémité d'une branche est appelée **hôte**, et correspond à une machine ou une entité du réseau. Le nom d'hôte qui lui est attribué doit être unique dans le domaine considéré, ou le cas échéant dans le sous-domaine. A titre d'exemple le serveur web d'un domaine porte ainsi généralement le nom *www*.

Le mot « **domaine** » correspond formellement au suffixe d'un nom de domaine, c'est-à-dire l'ensemble des étiquettes de noeuds d'une arborescence, à l'exception de l'hôte.

Le nom absolu correspondant à l'ensemble des étiquettes des noeuds d'une arborescence, séparées par des points, et terminé par un point final, est appelé
adresse FQDN (*Fully Qualified Domain Name*, soit *Nom de Domaine Totalement Qualifié*). La profondeur maximale de l'arborescence est de 127 niveaux et la longueur maximale d'un nom FQDN est de 255 caractères. L'adresse FQDN permet de repérer de façon unique une machine sur le réseau des réseaux. Ainsi *www.commentcamarche.net.* représente une adresse FQDN.

Les serveurs de noms

Les machines appelées *serveurs de nom de domaine* permettent d'établir la correspondance entre le nom de domaine et l'adresse IP des machines d'un réseau.

Chaque domaine possède un serveur de noms de domaines, appelé « serveur de noms primaire » (*primary domain name server*), ainsi qu'un serveur de noms secondaire (*secondary domaine name server*), permettant de prendre le relais du serveur de noms primaire en cas d'indisponibilité.

Chaque serveur de nom est déclaré dans à un serveur de nom de domaine de niveau immédiatement supérieur, ce qui permet implicitement une délégation d'autorité sur les domaines. Le système de nom est une architecture distribuée, où chaque entité est responsable de la gestion de son nom de domaine. Il n'existe donc pas d'organisme ayant à charge la gestion de l'ensemble des noms de domaines.

Les serveurs correspondant aux domaines de plus haut niveau **TLD** (*Top Level Domain*, soit *domaines de plus haut niveau*) sont appelés « **serveurs de noms racine** ». Il en existe treize, répartis sur la planète, possédant les noms « a.root-servers.net » à « m.root-servers.net ».

Un serveur de noms définit une zone, c'est-à-dire un ensemble de domaines sur lequel le serveur a autorité. Le système de *noms de domaine* est transparent pour l'utilisateur, néanmoins il ne faut pas oublier les points suivants :

- Chaque ordinateur doit être configuré avec l'adresse d'une machine capable de transformer n'importe quel nom en une adresse IP. Cette machine est appelée Domain Name Server. Pas de panique: lorsque vous vous connectez à internet, le fournisseur d'accès va automatiquement modifier vos paramètres réseau pour vous mettre à disposition ces serveurs de noms.
- L'adresse IP d'un second *Domain Name Server* (secondary Domain Name Server) doit également être définie : le serveur de noms secondaire peut relayer le serveur de noms primaire en cas de dysfonctionnement.

Le serveur le plus répandu s'appelle **BIND** (*Berkeley Internet Name Domain*). Il s'agit d'un logiciel libre disponible sous les systèmes UNIX, développé initialement par l'université de Berkeley en Californie et désormais maintenu par l'*ISC* (*Internet Systems Consortium*).

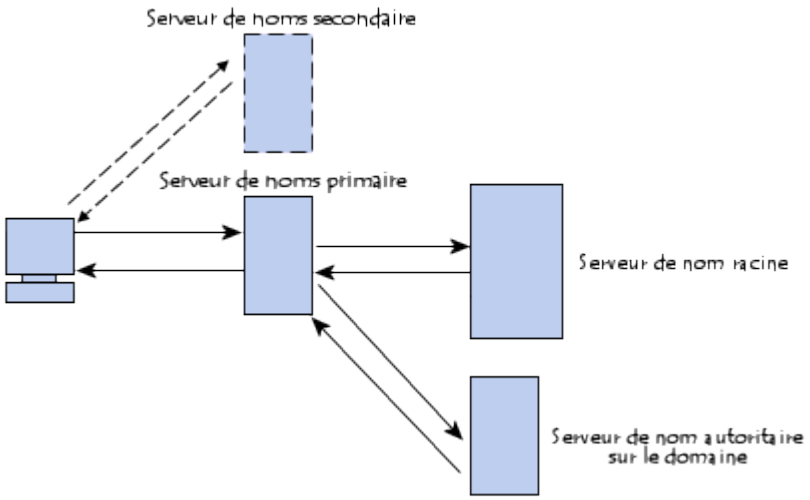
Résolution de noms de domaine

Le mécanisme consistant à trouver l'adresse IP correspondant au nom d'un hôte est appelé « **résolution de nom de domaine** ». L'application permettant de réaliser cette opération (généralement intégrée au système d'exploitation) est appelée « **résolveur** » (en anglais « *resolver* »).

Lorsqu'une application souhaite se connecter à un hôte connu par son nom de domaine (par exemple « www.commentcamarche.net »), celle-ci va interroger un serveur de noms défini dans sa configuration réseau. Chaque machine connectée au réseau possède en effet dans sa configuration les adresses IP de deux serveurs de noms de son fournisseur d'accès.

Une requête est ainsi envoyée au premier serveur de noms (appelé « serveur de nom primaire »). Si celui-ci possède l'enregistrement dans son cache, il l'envoie à l'application, dans le cas contraire il interroge un serveur racine (dans notre cas un serveur racine correspondant au TLD « .net »). Le serveur de nom racine renvoie une liste de serveurs de noms faisant autorité sur le domaine (dans le cas présent les adresses IP des serveurs de noms primaire et secondaire de *commentcamarche.net*).

Le serveur de noms primaire faisant autorité sur le domaine va alors être interrogé et retourner l'enregistrement correspondant à l'hôte sur le domaine (dans notre cas *www*).




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

REMARQUE: le programme libc peuvent ne pas supporter plus de 3 serveurs de noms.

Netstat

La commande Netstat permet de connaître les connexions TCP actives sur la machine sur laquelle elle est lancée et ainsi de lister l'ensemble des ports TCP et UDP ouverts sur l'ordinateur.

Vous pouvez voir 4 colonnes, la première indique le protocole (TCP, UDP), la seconde indique l'adresse locale ainsi que le port utilisé dans la connexion. La troisième indique l'adresse et le port destination et enfin, dans la dernière colonne, on voit l'état de la connexion.

l'option **-a** permet d'afficher toutes les connexions sur votre machine y compris les ports qui sont « à l'écoute » sur votre machine. Ainsi, si vous avez un serveur ftp qui tourne sur votre PC, vous devriez voir une entrée de ce type :

TCP nom_de_votre_pc:21 nom_de_votre_pc:0 LISTENING
L'option -n permet d'afficher les machines avec leur adresse IP et le services à l'aide du numéro de port plutôt que des informations au format texte.
Pour terminer, netstat offre une fonction très intéressante qui fourni la liste des routes actives sur votre PC au moyen de l'option -r.
netstat -r

```
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic   MSS Fenêtre irtt Iface
192.168.1.0     *               255.255.255.0   U         0 0          0 eth0
default         192.168.1.1     0.0.0.0         UG        0 0          0 eth0
```

netstat -s

```
Détail des statistiques

Ip:
    89510 total packets received
    0 forwarded
```

```
    0 incoming packets discarded
    88835 incoming packets delivered
    82291 requests sent out
Icmp:
    0 ICMP messages received
    0 input ICMP message failed.
    Histogramme d'entrée ICMP
    0 ICMP messages sent
    0 ICMP messages failed
    Histogramme de sortie ICMP
Tcp:
    844 active connections openings
    4 passive connection openings
    2 failed connection attempts
    18 connection resets received
    19 connections established
    87666 segments received
    80373 segments send out
    72 segments retransmited
    0 bad segments received.
    211 resets sent
Udp:
    1251 packets received
    0 packets to unknown port received.
    0 packet receive errors
    1926 packets sent
UdpLite:
TcpExt:
    348 TCP sockets finished time wait in fast timer
    2 time wait sockets recycled by time stamp
    245 delayed acks sent
    Quick ack mode was activated 42 times
    7 packets directly queued to recvmsg prequeue.
    3123 bytes directly in process context from backlog
    1342 bytes directly received in process context from prequeue
    70656 packet headers predicted
    28 packets header predicted and directly queued to user
    2199 acknowledgments not containing data payload received
    835 predicted acknowledgments
    10 times recovered from packet loss by selective acknowledgements
    3 congestion windows recovered without slow start after partial ack
    44 TCP data loss events
    41 fast retransmits
    31 other TCP timeouts
    42 DSACKs sent for old packets
    3 DSACKs received
    43 connections reset due to unexpected data
    10 connections reset due to early user close
    TCPSackShifted: 101
    TCPSackMerged: 65
    TCPSackShiftFallback: 35
IpExt:
    InMcastPkts: 46
    OutMcastPkts: 48
    InBcastPkts: 675
    OutBcastPkts: 675
    InOctets: 117058592
    OutOctets: 7977037
    InMcastOctets: 5467
    OutMcastOctets: 5547
    InBcastOctets: 95801
    OutBcastOctets: 95801
```

voir **Linux Attitude** : http://linux-attitude.fr/post/iproute2

ajouter une route

```
ip route add default via 10.0.0.1
```

allumer/ éteindre une carte réseau

```
ip link set dev eth0 up/down
```

arreter etho

```
ifdown eth0
```

Démarrer et arrêter les interfaces configurées

```
# /etc/init.d/networking stop
# /etc/init.d/networking start
# /etc/init.d/networking restart
```

forcer le mtu d'une carte

```
ip link set dev eth0 mtu 1492
```

forcer l'adresse mac d'une carte

```
ip link set dev eth0 address xx:xx
```

renommer une carte réseau

```
ip link set dev eth0 name toto
```

ajouter une ip (alias ip)

```
ip addr add 10.0.0.1/24 dev eth0
```

L'option + brd définit automatiquement l'adresse de diffusion tel que déterminé par le masque de réseau.

```
# ip address add 192.168.1.100/24 brd + dev eth0
```

Supprimer / Supprimer / Désactiver l'adresse IP d'une carte

```
ip address del 192.168.1.100 dev eth0
```

supprimer toutes les ip

```
ip addr flush dev eth0
```

vider la table arp

```
$ ip neigh flush
```

Linux: 20 Iptables Examples For New SysAdmins

http://www.cyberciti.biz/tips/linux-iptables-examples.html

Trouver l'adresse d'un site :

```
 host -t a www.facebook.com
www.facebook.com has address 69.171.242.13
```

Trouver l'adresse réseau correspondante

```
whois 69.171.228.40 | grep CIDR
CIDR:           69.171.224.0/19
```

Pour blquer un accès sortant à www.facebook.com, entrez:

```
# iptables -A OUTPUT -p tcp -d 69.171.224.0/19 -j DROP
```

Trouver qui écoute le port 80

```
 netstat -tulpn|grep 80
tcp6       0      0 :::80                   :::*                    LISTEN
2198/apache2
```

Et le port 21

```
netstat -tulpn|grep 21
tcp        0      0 0.0.0.0:21              0.0.0.0:*               LISTEN
3058/vsftpd
tcp6       0      0 :::80                   :::*                    LISTEN
2198/apache2 (petite erreur d'interprétation)
```

Liste des commande de réseau de bas niveau

commande	description
`ifconfig`	afficher l'état et l'adresse du lien des interfaces actives
`ip addr show`	afficher l'état et l'adresse du lien des interfaces actives
`route -n`	afficher toutes les tables de routage sous forme d'adresses numériques
`ip route show`	afficher toutes les tables de routage sous forme d'adresses numériques
`arp`	afficher le contenu actuel des tables de cache d'ARP
`ip neigh`	afficher le contenu actuel des tables de cache d'ARP
`plog`	afficher le journal du démon ppp
`ping yahoo.com`	vérifier la connexion internet vers « `yahoo.com` »
`whois yahoo.com`	vérifier qui a enregistré « `yahoo.com` » dans la base de données des domaines
`traceroute yahoo.com`	tracer la connexion Internet vers « `yahoo.com` »
`tracepath yahoo.com`	tracer la connexion Internet vers « `yahoo.com` »
`mtr yahoo.com`	tracer la connexion Internet vers « `yahoo.com` » (de manière répétitive)
`dig [@dns-serveur.com] example.com [{a|mx|any}]`	vérifier les enregistrements DNS de « `example.com` » par « `dns-serveur.com` » pour un enregistrement « `a` », « `mx` » ou « `any` »
`iptables -L -n`	vérifier le filtre de paquets
`netstat -a`	rechercher tous les ports ouverts
`netstat -l --inet`	rechercher les ports à l'écoute
`netstat -ln --tcp`	rechercher les ports TCP à l'écoute (numérique)
`dlint example.com`	vérifier les information de zone DNS de « `example.com` »

Ancienne méthode de configuration et connexion réseau

L'interface loopback

Maintenant, nous allons configurer votre réseau. Une fois que vous êtes en ***root*** :

Sur la plupart des distributions Linux, l'interface loopback est deja configurée. Vous pouvez le verifier en faisant la commande suivante :

```
# /sbin/ifconfig
```

Vous devriez voir ceci :

```
# ifconfig lo
lo              Link encap Local loopback
                 inet addr 127.0.0.1 Bcast [NONE SET] Mask 255.0.0.0
                 UP BROADCAST LOOPBACK RUNNING MTU 2000 Metric 1
                 RX packets 0 errors 0 dropped 0 overrun 0
                 TX packets 0 errors 0 dropped 0 overrun 0
```

sinon faites :

```
# ifconfig lo 127.0.0.1
```

Il faut maintenant entrer cette interface dans la table de routage

```
# route add 127.0.0.1
```

Maintenant que votre interface loopback est configuré il suffit de tester votre interface en faisant des "ping". Nous verrons le fonctionnement de "ping" dans la section des outils réseaux.

```
# ping 127.0.0.1
PING localhost (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=32 time=1 ms
64 bytes from 127.0.0.1: icmp_seq=0 ttl=32 time=0 ms
```

Votre interface loopback est correctement configurée.

L'interface ethernet

La configuration de l'interface Ethernet utilise les memes outils et les memes méthodes que l'interface Loopback.
Nous allons configurer cette interface avec une classe C pour 254 clients. Mais vous pouvez changer l'adressage ip en suivant le tableau cité plus haut. Avant d'effectuer ceci, il faut avoir inseré le module réseau correspondant à votre carte.

```
# ifconfig eth0 192.168.0.1 netmask 255.255.255.0
```

en faisant

```
# /sbin/ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:01:03:48:77:56
                inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
                UP BROADCAST RUNNING  MTU:1500  Metric:1
                RX packets:0 errors:0 dropped:0 overruns:1 frame:0
                TX packets:0 errors:0 dropped:0 overruns:0 carrier:4
                collisions:0 txqueuelen:0
                Interrupt:11 Base address:0x1000
```

Il faut ajouter l'interface ethernet dans la table de routage :

```
# route add -net 192.168.0.0
```

Vous devez deja voir l'interface "*lo*" aussi connu sous le nom de loopback sinon *ifconfig lo 127.0.0.1*

```
# ping 192.168.0.1
1 packets transmitted, 1 packets received, 0% packet loss
rOund-trip min/avg/max = 0.1/0.1/0.1 ms
bash-2.04$ ping 192.168.0.1
PING 192.168.1.223 (192.168.0.1): 56 data bytes
64 bytes from 192.168.0.1: icmp_seq=0 ttl=128 time=0.5 ms
64 bytes from 192.168.0.1: icmp_seq=1 ttl=128 time=0.3 ms
--- 192.168.0.1 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.3/0.4/0.5 ms
```

Vos interfaces sont désormais configurées correctement.
Vous pouvez maintenant editer votre fichiers */etc/hosts* :

```
#Debut du fichier /etc/hosts.
127.0.0.1 localhost
192.168.0.1 Albert.einstein.net Albert
#Fin du fichier /etc/hosts.
```

Essayez de faire

```
# ping localhost
# ping Albert
```

Par Rémy Pouchain