

Linux - La couche réseau

Les périphériques

- **eth0, eth1, ...** : périphériques ethernet
- **tr0, tr1, ...** : périphériques TokenRing
- **ppp0, ppp1, ...** : les interfaces PPP (Point-to-Point Protocol : liaisons commutées)
- **lo** est le périphérique de loopback

Le paramétrage

Il peut être fait en éditant les fichiers paramètres.

Les fichiers de configuration

Les fichiers permettant de configurer la couche réseau sont les suivants (sur les distributions RedHat/Fedora/Mandriva) :

- **/etc/hosts** (associations noms de machines ↔ adresses IP)
- **/etc/networks** (associations noms de réseaux ↔ adresses IP)
- **/etc/sysconfig/network** (lance le réseau)
- **/etc/resolv.conf** (configuration du résolveur DNS)
- **/etc/host.conf** (décrit la méthode de résolution)
- **/etc/init.d/networking** (active les interfaces Ethernet configurées au démarrage ensemble de fichiers dans **/etc/network/interfaces**. Ils permettent la configuration des interfaces et permettent de contrôler et d'obtenir l'état des liaisons.

Le fichier /etc/hosts

Il permet d'effectuer la résolution de noms en l'absence de serveur DNS. Il comporte l'adresse IP suivie du nom complet de la machine puis éventuellement d'un alias ou nom court. Chaque information est séparée des précédentes par un ou plusieurs espaces .

Exemple de fichier **/etc/hosts** :

```
# /etc/hosts

127.0.0.1      localhost.localdomain localhost    # boucle locale (loopback)

192.168.1.1   yoda.darkstar.org yoda
192.168.1.2   luke.darkstar.org luke
```

Le fichier /etc/host.conf

Il permet de spécifier la méthode de résolution de noms : fichiers hosts, serveur DNS ou NIS et d'indiquer l'ordre de résolution.

Pour plus d'information, taper `man host.conf`

Exemple de fichier **/etc/host.conf**

```
order hosts,bind      # resolution d'abord par le fichier hosts puis par bind
multi on              # machines multi-homes
```

Le fichier /etc/resolv.conf

indique le nom de domaine de l'ordinateur, l'adresse du ou des serveurs DNS (s'ils existent)

```
domain mondomaine.fr
nameserver 195.98.246.50 # adresse du serveur primaire DNS (jusqu'à 3 serveurs)
search autredomaine.fr # liste de domaines à essayer, pour les noms d'hôte non FQDN
1
```

Le répertoire `/etc/network/interfaces`

```
iface eth0 inet static
address 192.168.1.5
netmask 255.255.255.0
gateway 192.168.1.254
```

pour DHCP

```
auto eth0
iface eth0 inet dhcp
```

voir man page de dhcp

Ifconfig ou ip (Debian Stretch)

Ifconfig permet de configurer une interface ou d'afficher son paramétrage. ifconfig donne la liste des périphériques actifs ainsi que leurs paramètres

Ajouter et retirer des interfaces

La commande ifconfig permet de configurer des interfaces. La syntaxe est :

```
ifconfig -a
ifconfig eth0 xxx.xxx.xxx.xxx ou ip addr add 192.168.1.5/24 dev enp0s3
ifconfig eth0 up ou ip link set enp0s3 up
ifconfig eth0 down ou ip link set enp0s3 down
```

Attention : la commande **ifconfig** avec des paramètres permet de changer les paramètres d'une interface, mais ceux-ci ne seront pas conservés lors d'une nouvelle configuration de l'interface

La commande **ifconfig** ou **ip a** sans argument équivaut à la première : elle affiche la liste des interfaces actives du système.

Exemple de sortie de la commande ifconfig :

```
eth0 : Lien encap:Ethernet HWaddr 00:40:95:36:70:37
inet adr:192.168.13.30 Bcast:192.168.13.255 Masque:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
Paquets Reçus:1686 erreurs:0 jetés:0 débordements:0 trames:1
Paquets transmis:4098 erreurs:0 jetés:0 débordements:0 carrier:0
collisions:0 lg file transmission:100
Interruption:11 Adresse de base:0x6000 10 Lien encap:Boucle locale

lo : inet adr:127.0.0.1 Masque:255.0.0.0
UP LOOPBACK RUNNING MTU:3924 Metric:1
Paquets Reçus:722 erreurs:0 jetés:0 débordements:0 trames:0
Paquets transmis:722 erreurs:0 jetés:0 débordements:0 carrier:0
collisions:0 lg file transmission:0
```

*La seconde permet d'attribuer une adresse IP à une interface donnée. C'est ce que fait l'outil de configuration du réseau lors de l'installation de la machine, ou l'outil de configuration réseau de **Linuxconf** . Généralement, l'adresse IP est le seul paramètre nécessaire, mais, si l'on se trouve sur un sous-réseau, il peut être nécessaire de préciser le masque correspondant.*

Les deux dernières commandes permettent respectivement d'activer et de désactiver une interface du système.

Attribuer plusieurs adresses à une seule interface

Il est possible d'attribuer plusieurs adresses IP à une même interface. Cela s'appelle l'**IP aliasing**. On peut vouloir l'utiliser pour :

- Héberger plusieurs serveurs d'un même type sur une même machine physique (deux serveurs HTTP distincts, par exemple) ;
- Insérer rapidement un serveur de secours dans le réseau local, en cas de défaillance du premier...

L'**IP aliasing** exige l'intégration d'un support spécifique au niveau du noyau. La plupart des noyaux Linux sont configurés pour supporter cette fonctionnalité, et ne nécessitent donc pas d'autre modification.

Ceci fait, on peut configurer une interface pour utiliser plusieurs adresses IP :

```
ifconfig interface:alias adresse.ip.numerique
ifconfig eth0:0 192.168.2.2
ifconfig eth0:1 192.168.2.3
```

La même carte réseau **eth0** est configurée par les deux dernières lignes de l'exemple pour utiliser les adresses IP

192.168.2.3 et **192.168.2.2**.

Note : La configuration réseau de la carte est stockée dans les fichiers :

/etc/sysconfig/network : paramètres génériques ;

/etc/network-scripts/ifcfg-eth0 : paramètres de l'interface réseau eth0.

Netstat

Permet d'afficher les statistiques TCP/IP . cf les manpages pour obtenir les différentes options.

Exemple :

```
[phil@phoenix phil]$ netstat -ta
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State
tcp 0 40 phoenix.syroidmanor:ssh 192.168.1.5:1132 ESTABLISHED
tcp 0 0 *:ssh *:* LISTEN
tcp 0 0 phoenix.syroidmano:1028 hydras.syro:netbios-ssn ESTABLISHED
tcp 0 0 phoenix.syroidmano:1027 raidserver:netbios-ssn ESTABLISHED
tcp 0 0 *:printer *:* LISTEN
tcp 0 0 *:auth *:* LISTEN
tcp 0 0 *:1024 *:* LISTEN
tcp 0 0 *:sunrpc *:* LISTEN
```

Le routage

Pour activer le routage, choisira activer le routage dans **Linuxconf** ou **Webmin**

La commande **route** permet d'afficher et de définir des routes :

Ajouter ou effacer une route

route add -host xxx.xxx.xxx.xxx eth0 ou **ip route add**

route add -net xxx.xxx.xxx.xxx eth1 ou **ip route add 192.168.1.0/24 via 172.16.0.1**

route del -host xxx.xxx.xxx.xxx eth0 ou **ip route del 192.168.1.0/24 via 172.16.0.1**

route -n ou ip r

```
[root@linux phil] # netstat -rn
Kernel IP routing table
Destination      Gateway          Genmask         Flags   MSS Window  irtt Iface
192.168.1.0      0.0.0.0         255.255.255.0  U       0  0        0 eth0
127.0.0.0        0.0.0.0         255.0.0.0      U       0  0        0 lo
0.0.0.0          192.168.1.1    0.0.0.0        UG      0  0        0 eth0
```

La commande **route** permet d'associer soit une adresse IP directement, soit un ensemble d'adresses, à une interface donnée.

Ainsi, **route add -net 192.168.1.0 eth1** associe la deuxième interface réseau à l'ensemble des machines dont l'adresse IP est **192.168.1.***. En cas de conflit, le système utilise la première route trouvée.

Déclarer une route vers une passerelle

Une passerelle est une machine qui fait le lien entre deux réseaux distincts. Dans le cas d'un réseau local, l'application la plus fréquente est une passerelle connectée à Internet.

La passerelle dispose de méthodes plus évoluées pour router les paquets. Il est recommandé de définir une **route par défaut**, utilisée par tous les paquets, et pointant vers la passerelle.

route add default gw adresse.de.la.passerelle ou **ip route add default via 192.168.1.1**

La passerelle est chargée d'acheminer les paquets vers leur destination finale.

Quelques commandes utiles pour la gestion réseau

tcpdump : affiche les en-têtes de paquet sur une interface.

ex : **tcpdump -i eth1** # affiche tous les paquets qui arrivent sur l'interface eth1
tcpdump host yoda port 80 # affiche tous les paquets envoyés ou recus par la machine yoda sur le port 80

tcpdump permet également de capturer le contenu des paquets

traceroute : sans commentaire

arp : permet de voir et modifier le contenu des tables ARP

nc : network cat

Ex : **nc 192.168.1.1 80** # se connecte à la machine sur le port 80

nmap : scanner des ports

Ex : **nmap 192.168.1.1** #affiche les ports ouverts sur la machine en question

Ex : **nmap "192.168.1.*"** #scanne toutes les adresses du réseau

Voir également **nmapfe** (front-end graphique de nmap)

arpwatch : démon permettant de garder des statistiques ARP (Nom machine, adresse IP, adresse MAC)

il existe aussi pour la gestion réseau les utilitaires suivants : **iptraf** qui donne des statistiques sur les interfaces, **wireshark/ethereal** (sous X Window, fonctionne comme le moniteur réseau Microsoft), **ntop** : permet d'obtenir des statistiques réseau .

Pour effectuer du routage dynamique avec **RIP**, on utilisera le démon **routed**, **gated** permettant quant à lui d'utiliser **OSPF**.