

MODULE 3

Gestion du service réseau

Objectifs de ce module :

- ✓ **Connaître les fichiers liés à ce service**
- ✓ **Utiliser les commandes essentielles**
- ✓ **Activer le réseau manuellement**

Table des matières

Sujets	Page
Introduction.....	3
Les interfaces réseaux.....	4
Éléments de configuration.....	5
Configuration statique.....	6
Configuration dynamique.....	6
Les fichiers de configuration relié au réseau.....	7
La résolution de nom.....	7
Avec le fichier nsswitch.conf.....	7
Avec le fichier /etc/hosts.....	7
nsswitch.conf ou host.conf ?.....	7
Le fichier /etc/networks.....	8
Le fichier /etc/resolv.conf.....	8
Les fichiers de configuration des interfaces réseau.....	9
Préparation.....	9
Dossier des interfaces réseaux.....	9
Configuration des interfaces.....	11
Cas d'une interface câblée avec adresse IP statique.....	11
Cas d'une interface câblée avec adresse IP reçue d'un DHCP.....	12
Cas d'une interface sans-fil.....	12
Passerelle par défaut.....	13
Les outils de l'administrateur réseau.....	14
La commande ifconfig.....	14
La commande route.....	17
Ajout d'une route.....	19
Suppression d'une route.....	19
La commande netstat.....	20
La commande traceroute.....	22
La commande dig.....	23

Introduction

Le service réseau est habituellement géré par un service qui se nomme "NetworkManager". C'est un service de contrôle du réseau dynamique qui permet de rendre les connexions actives et disponibles lorsqu'elles deviennent disponibles. Le "NetworkManager" consiste en un applet qui se loge dans le tableau de notification de la barre des tâches. Il permet d'afficher l'information reliée aux cartes réseaux de même que de configurer les cartes, les modifier ou créer des nouvelles connexions réseaux. NetworkManager peut être utilisé pour configurer les types de connexions suivantes:

- Ethernet
- sans-fil
- connexions mobiles (3G, 4G)
- DSL
- PPPoE

Il peut créer des alias réseaux, des routes statiques, des informations pour le DNS et des connexions VPN.

C'est un service qui se veut convivial et qui trouve toute son utilité lorsque votre machine se déplace d'une place à une autre. C'est ce qui fait que le service « NetworkManager » est très intéressant pour les ordinateurs (clients) portables (laptop) qui se déplace d'un wifi à un autre. NetworkManager améliore ainsi l'expérience usager quant à la connectivité à ce genre de réseau.



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.

Dans le cas d'ordinateurs qu'on utilisera comme serveur, on aime plutôt que les adresses IP ne changent pas et soient plutôt statiques. Dans ce cas, on préférera plus prendre le contrôle nous-même de la gestion du réseau avec un service de réseau qui se nomme « networking ».

Les interfaces réseaux

Les réseaux informatiques utilisent un modèle composé de plusieurs couches de protocoles. Nous nous intéressons ici à la troisième couche, dite *couche réseau*, qui utilise le protocole IP (Internet protocol), dans sa version 4 ou 6 : c'est cette couche qui définit la topologie des réseaux, et dont la configuration est par conséquent très importante.

Du point de vue du système d'exploitation, ce périphérique est une *interface réseau*. Sous Linux, ces interfaces sont nommées de différentes façon dépendamment sur quel port d'extension votre carte branchée.

Dans certains systèmes de type RedHat les cartes réseaux ont des appellations différentes de celles utilisées par un système de type Debian.

Appellation des interfaces réseaux (ancienne)

Type d'interface	Nom
Ethernet	eth0 (1ère carte), eth1(2ème carte), etc.
Sans-fil	wlan0 (1ère carte), wlan1(2ème carte)

Cette appellation a été abandonné pour les raisons que l'on cite ici

«Comme l'exploration du pilote n'est généralement pas prévisible pour la technologie moderne, cela signifie que dès que plusieurs interfaces réseau sont disponibles, l'attribution des noms "eth0", "eth1" et ainsi de suite n'est généralement plus fixe et il se peut très bien que " eth0 "sur un démarrage finit par être" eth1 "sur le suivant. Cela peut avoir de graves implications pour la sécurité, par exemple dans les règles de pare-feu qui sont codées pour certains schémas de dénomination, et qui sont donc très sensibles aux changements de noms imprévisibles.»

Avec la nouvelle appellation des interfaces, vous êtes assuré, entre autres, de :

- Noms d'interface stables lors des redémarrages
- Noms d'interface stables même lorsque du matériel est ajouté ou supprimé, c'est-à-dire qu'aucune ré-énumération n'a lieu
- Noms d'interface stables lorsque les noyaux ou les pilotes sont mis à jour / modifiés
- Noms d'interface stables même si vous devez remplacer les cartes Ethernet cassées par de nouvelles
- La même chose sur toutes les distributions qui ont adopté systemd.

Éléments de configuration

Une configuration réseau complète, permettant de profiter d'un réseau ou de l'Internet, est constituée des éléments suivants :

une adresse IP

cette adresse identifie votre *hôte* sur le réseau où il est connecté ;

un masque de sous-réseau

cette donnée indique la partie de votre adresse qui caractérise le réseau local sur lequel votre hôte est connecté, et lui permet de déterminer, pour n'importe quelle adresse IP, si celle-ci fait ou non partie du réseau local ;

une passerelle par défaut

c'est l'adresse IP à laquelle il faut transmettre les paquets IP destinés à des hôtes situés hors du réseau local, pour qu'ils soient *routés* vers le réseau local de leur destinataire ;

des serveurs DNS

ce sont les adresses de serveurs auxquels votre système ira demander les correspondances entre noms de domaine (*www.linuxmint.com*) et adresses IP (192.124.249.53).

Chaque élément de configuration est nécessaire pour pouvoir utiliser normalement le réseau ou l'Internet :

- sans adresse IP, il est impossible de recevoir les réponses à ses requêtes ;
- sans masque de sous-réseau ou sans passerelle par défaut, il est impossible de communiquer avec les hôtes situés hors du réseau local ;
- sans serveur DNS, on ne peut pas désigner un hôte par son nom de domaine, et il faut donc connaître les adresses IP de tous les serveur que l'on souhaite utiliser

Masque de sous-réseau

Le sous-réseau désigne votre réseau local ou LAN (*local area network*). Il est défini par un *préfixe* d'adresse, par exemple 192.168.0 : toutes les adresses IP qui commencent par ce préfixe font partie de votre réseau local. Il peut être écrit de deux façon :

par sa longueur

en nombre de bits, notée **/longueur** : dans notre exemple, **/24** (chaque chiffre d'une adresse IP fait un octet, soit 8 bits) ;

par un masque

semblable à une adresse IP dont tous les bits sont à 1 dans la partie correspondant au préfixe, et à 0 dans la partie restante : dans notre exemple, 255.255.255.0.

Configuration statique

C'est le mode de configuration le plus simple à comprendre : vous devez connaître à l'avance votre configuration complète, pour l'appliquer sur votre système.

Configurer votre connexion consiste alors à affecter à votre carte réseau son adresse IP et son masque de sous-réseau, à ajouter la passerelle par défaut à la table de routage du noyau Linux, et à noter l'adresse des serveurs DNS dans le fichier de configuration du *résolveur DNS*.

Configuration dynamique

Ce mode de configuration, désormais très répandu, est plus adapté aux ordinateurs portables, susceptibles d'être connectés à des réseaux différents, ou aux gens qui ne veulent pas avoir besoin de configurer eux-même leur connexion.

Pour cela, lorsque votre système démarre, ou détecte qu'il vient d'être connecté à un réseau, envoie une demande de paramètres de connexion. Pour un réseau IPv4, cette demande utilise le protocole DHCP (dynamic host configuration protocol) ; pour un réseau IPv6, elle s'effectue dans le cadre d'un processus appelé *découverte de voisinage*, ou par DHCPv6.

Sur un réseau permettant les configurations dynamiques, un serveur répond alors en vous proposant une configuration, qui est alors appliquée sur votre système.

Les fichiers de configuration relié au réseau

La résolution de nom

Avec le fichier nsswitch.conf

La résolution de nom est maintenant assuré par un fichier qui se nomme « nsswitch.conf » situé dans le dossier /etc.

Ce fichier est habituellement dans le dossier /etc

La ligne « hosts :... » indique la séquence qui est vérifié pour assurer la résolution de nom.

```
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the 'glibc-doc-reference' and 'info' packages installed, try:
# 'info libc "Name Service Switch"' for information about this file.
passwd:         files systemd
group:          files systemd
shadow:         files
gshadow:        files

hosts:          files mdns4_minimal [NOTFOUND=return] dns myhostname
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

Figure 1: Contenu du fichier nsswitch.conf

files : signifie que le système ira vérifier le fichier /etc/hosts en premier

mdns4_minimal [NOTFOUND=return] : multicast dns pour ipv4 et les noms se terminant en .local. Le NOTFOUND indique que si on en trouve pas, on arrête immédiatement la lecture de cette ligne. On ne se rendra donc pas à dns.

Pour un serveur, je vous conseille donc plus de modifier la ligne hosts comme ceci :

```
hosts : files  dns
```

Avec le fichier /etc/hosts

Le fichier `hosts` donne un moyen d'assurer la résolution de noms, de donner un nom FQDN à un hôte.

Exemple de fichier `hosts`

```
127.0.0.1 localhost localhost.localdomain
192.168.1.1 ordi1
192.168.1.2 ordi2
```

nsswitch.conf ou host.conf ?

Depuis quelques années les 2 fichiers sont présents sur les systèmes. De plus en plus, nsswitch.conf est la méthode privilégiée. Le fichier /etc/host.conf est encore là par soucis de compatibilité.

Le fichier `/etc/networks`

Il permet d'affecter un nom logique à un **réseau**

```
localnet 127.0.0.0  
foo-net 192.168.1.0
```

Cette option permet par exemple d'adresser un réseau sur son nom, plutôt que sur son adresse.

route add foo-net au lieu de **route add -net 192.168.1.0**.

Le fichier `/etc/resolv.conf`

Il permet d'affecter les serveurs de noms (dns) à utiliser.

Exemple

```
nameserver 192.168.1.1  
nameserver 8.8.8.8
```

Ici le fichier déclare 2 machines chargées de la résolution de noms.

Les fichiers de configuration des interfaces réseau

Préparation

Premièrement, désactivez le service “NetworkManager”. Nous utiliserons le service “networking” pour configurer nous même nos interfaces réseau.

```
systemctl stop NetworkManager  
systemctl disable NetworkManager
```

Installer le paquet resolvconf

```
apt install resolvconf
```

Resolvconf est reliée à la résolution de nom.

Dossier des interfaces réseaux

Vous trouverez ces fichiers dans `/etc/network/`.

Le fichier `interfaces` contient les informations des cartes réseaux. Il est conseillé de laisser ce fichier tel quel et d’ajouter plutôt les informations reliées aux interfaces dans des fichiers que l’on ajoutera dans le dossier “`interfaces.d`”.

Le fichier “`interfaces`” va lire le contenu du dossier `interfaces.d` avec la commande “`source-directory /etc/network/interfaces.d`” et applique toutes les informations trouvées dans chaque fichier qui s’y trouve.

Il s’agit de créer les fichiers des interfaces que l’on veut contrôler en leur donnant un nom qui commence par `ifcfg-` suivi du nom de l’interface donné par le système.

La commande ifconfig avec l'option -a vous permettra de faire afficher les interfaces réseaux.

On distingue ici 3 interfaces réseaux. Le nom de ces interfaces sont affichés dans la colonne de gauche (le nom avant le :)

```
schasse@HawkLmg:/etc/network$ ifconfig -a
enp0s31f6: flags=4098<BROADCAST,MULTICAST> mtu 1500
    ether 04:0e:3c:44:41:2d txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 16 memory 0xf0500000-f0520000

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Boucle locale)
    RX packets 3515 bytes 714615 (714.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3515 bytes 714615 (714.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.200 netmask 255.255.255.0 broadcast 10.0.0.255
    inet6 fe80::5e87:9cff:fe4d:7605 prefixlen 64 scopeid 0x20<link>
    inet6 2607:fa49:d93e:6400:5e87:9cff:fe4d:7605 prefixlen 64 scopeid 0x0<global>
    inet6 2607:fa49:d93e:6400:b15e:bb6b:b0bd:6a8a prefixlen 64 scopeid 0x0<global>
    ether 5c:87:9c:4d:76:05 txqueuelen 1000 (Ethernet)
    RX packets 141467 bytes 176791469 (176.7 MB)
    RX errors 0 dropped 2 overruns 0 frame 0
    TX packets 36722 bytes 8852096 (8.8 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Dans la prise d'écran ci-dessus, on aperçoit:

enp0s31f6 : La carte ethernet câblée

lo: Le loopback (127.0.0.1)

wlp1s0: La carte sans-fil.

Ça semble compliqué mais en réalité, il s'agit de regarder la première lettre pour constater que la lettre "e" est reliée aux interfaces câblées (ethernet), les lettres "wl" aux interfaces sans-fil (wl: wireless) et le lo relié au loopback.

Configuration des interfaces

Cas d'une interface câblée avec adresse IP statique

Nous allons définir les informations réseau pour une carte câblée nommée enp0s8

- Allez dans le dossier /etc/network/interfaces.d
- Créer le fichier ifcfg-enp0s8 avec la commande : `nano ifcfg-enp0s8`
(Évidemment, utilisez le nom associé à votre interface sur votre ordinateur)

Dans ce fichier, entrez les éléments suivants:

```
auto enp0s8
iface enp0s8 inet static
address 192.168.1.1
netmask 255.255.255.0
broadcast 192.168.1.255

#gateway adresse à déterminer
#dns-nameservers à déterminer
# Une ligne commençant par # est un commentaire
```

- Sauvegardez votre fichier
- Redémarrez le service réseau : `systemctl restart networking`

Vérifiez que votre interface réseau nommé enp0s8 à bien la bonne adresse.

```
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.1.1 netmask 255.255.255.0 broadcast 192.168.1.255
inet6 fe80::a00:27ff:feb7:81da prefixlen 64 scopeid 0x20<link>
ether 08:00:27:b7:81:da txqueuelen 1000 (Ethernet)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 32 bytes 4199 (4.1 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Cas d'une interface câblée avec adresse IP reçue d'un DHCP

Ce cas est très simple puisque vous n'avez qu'à indiquer que l'interface va recevoir son adresse d'un serveur DHCP en remplaçant static par dhcp comme ci-dessous:

```
auto enp0s8
iface enp0s8 inet dhcp
```

- Sauvegardez votre fichier
- Redémarrez le service réseau : `systemctl restart networking`

On a évidemment pas besoin des lignes supplémentaires indiquant l'adresse, le masque et le broadcast puisque toutes ces informations seront reçues du serveur DHCP.

Cas d'une interface sans-fil

Que ce soit une adresse reçue par DHCP ou statique, les informations sont les mêmes dépendamment si vous avez choisi un cas ou l'autre. Les seuls éléments à ajouter sont les informations pour le SSID de votre réseau sans-fil de même que le mot de passe encryptée de votre réseau sans-fil pour un réseau avec sécurité de type WPA2.

Voici un exemple pour une carte sans-fil dont le fichier se nomme `ifcfg-wlp1s0` :

```
auto wlp1s0
iface wlp1s0 inet static
address 10.0.0.100
netmask 255.255.255.0
broadcast 10.0.0.255
gateway adresse_ici
dns-nameservers 8.8.8.8 8.8.4.4
wpa-ssid le_nom_de_votre_reseau_ssid_sans-fil
wpa-psk le mot de passe encrypté ici
```

Le mot de passe encrypté peut-être générer de la façon suivante:

```
wpa_passphrase le_nom_ssid_ici mot_de_passe_de_votre_reseau
```

Voici un exemple:

```
schasse@Hawking:/etc/network/interfaces.d$ wpa_passphrase ssid mot_de_passe
network={
    ssid="ssid"
    #psk="mot_de_passe"
    psk=2bdc36041ad8a6221393a0fbbd381ac0ec9c90e5c8b0dd245500c98f52a24b5f
}
```

Passerelle par défaut

La passerelle par défaut permet de définir l'hôte de votre réseau local vers lequel envoyer tous les paquets destinés à des hôtes situés hors du réseau local, ce qui définit une *route par défaut*. Cette route doit être ajoutée à la table de routage du noyau Linux, avec la commande **route**. Par exemple, si votre passerelle par défaut a pour adresse IPv4 192.168.0.1, et peut être jointe par votre interface *eth0*, ajoutez ainsi la route par défaut :

```
route add default gw 192.168.0.1 dev eth0
```

Si vous êtes sur un réseau IPv6 qui possède une passerelle 2001:db8::1, utilisez l'option **--inet6**, qui permet de manipuler la table de routage IPv6 plutôt que la table IPv4 :

```
route --inet6 add default gw 2001:db8::1 dev eth0
```

Pour afficher la table de routage complète, utilisez la commande **route** sans arguments. Pour supprimer une route, utilisez la commande **route del**, suivie des mêmes arguments :

```
% route --inet6
Table de routage IPv6 du noyau
Destination                Next Hop                    Flag Met Ref Use If
2001::/32                   ::                          U    256 0    1 eth0
fe80::/64                   ::                          U    256 0    0 eth0
::/0                        2001:db8::1                UG   1    0    0 eth0
ff00::/8                    ::                          U    256 0    0 eth0
# route --inet6 del default gw 2001:db8::1 dev eth0
```

Note

La configuration IP et la table de routage peuvent être manipulées de façon plus avancée par une commande unique, **ip**, disponible dans le paquet *iproute*, dont vous pouvez consulter le [site web](http://www.linux-foundation.org/en/Net:Iproute2) à l'adresse <http://www.linux-foundation.org/en/Net:Iproute2>.

Les outils de l'administrateur réseau

La commande `ifconfig`

La commande `ifconfig` permet la configuration locale ou à distance des interfaces réseau de tous types d'équipements (unité centrale, routeur). Sans paramètres, la commande `ifconfig` permet d'afficher les paramètres réseau des interfaces.

La ligne de commande est :

```
ifconfig interface adresse [parametres].
```

Exemple :

`ifconfig enp0s3`

Renvoie les informations sur l'interface `enp0s3`

```
schasse@TuxServeur:~$ ifconfig enp0s3
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
inet6 fe80::2435:4765:74d0:107a prefixlen 64 scopeid 0x20<link>
ether 08:00:27:97:14:04 txqueuelen 1000 (Ethernet)
RX packets 203 bytes 28748 (28.7 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 250 bytes 26032 (26.0 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Voici les principaux arguments utilisés :

Nom du paramètre	Description
<code>enp0s3</code> , <code>enp0s8</code> , <code>wlp1s0</code> , ...	l'interface logique ou physique, il est obligatoire,
<code>up</code>	Active l'interface
<code>down</code>	Désactive l'interface
<code>netmask</code>	affecter un masque de sous-réseau
<code>broadcast</code>	définit l'adresse de broadcast
<code>arp</code> ou <code>-arp</code>	active ou désactive l'utilisation du cache arp de l'interface
<code>metric</code>	paramètre utilisé pour l'établissement des routes dynamiques, et déterminer le " coût " (nombre de sauts ou " hops ") d'un chemin par le protocole RIP.

multicast	active ou non la communication avec des machines qui sont hors du réseau.
promisc ou -promisc	activer ou désactiver le mode promiscuité de l'interface. En mode <i>promiscuous</i> , tous les paquets qui transitent sur le réseau sont reçus également par l'interface. Cela permet de mettre en place un analyseur de trame ou de protocole.

En faisant la commande `ifconfig`, nous obtenons:

```

schasse@TuxServeur:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
  inet6 fe80::2435:4765:74d0:107a prefixlen 64 scopeid 0x20<link>
  ether 08:00:27:97:14:04 txqueuelen 1000 (Ethernet)
  RX packets 203 bytes 28748 (28.7 KB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 250 bytes 26032 (26.0 KB)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  ether 08:00:27:b7:81:da txqueuelen 1000 (Ethernet)
  RX packets 0 bytes 0 (0.0 B)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 271 bytes 43558 (43.5 KB)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
  inet 127.0.0.1 netmask 255.0.0.0
  inet6 ::1 prefixlen 128 scopeid 0x10<host>
  loop txqueuelen 1000 (Boucle locale)
  RX packets 213 bytes 23143 (23.1 KB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 213 bytes 23143 (23.1 KB)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Analysons la sortie pour l'interface `enp0s3` :

```
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::2435:4765:74d0:107a prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:97:14:04 txqueuelen 1000 (Ethernet)
    RX packets 203 bytes 28748 (28.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 250 bytes 26032 (26.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

`enp0s3` : Nom de l'interface
UP : L'interface est active
BROADCAST et MULTICAST sont acceptés.

inet : Adresse IP (v4) associée à l'interface
netmask et broadcast vont de soi.

inet6 : adresse IP (v6) de l'interface si associée.
ether : L'adresse MAC
RX : Les paquets transmis avec les paquets en erreurs et ceux complètement laisser tombés
TX : Les paquets transmis avec les même informations que pour RX.

Si une des ces lignes n'est pas présente, c'est que l'interface n'a pas cette information active. Si la ligne `inet` n'est pas affichée, c'est que cette interface n'a pas d'adresse IP qui lui a été assignée et donc l'interface est probablement inactive.

Voici la séquence de commande pour démontrer tout cela:

```
schasse@TuxServeur:~$ sudo ifconfig enp0s3 down
schasse@TuxServeur:~$ ifconfig enp0s3
enp0s3: flags=4098<BROADCAST,MULTICAST> mtu 1500
    ether 08:00:27:97:14:04 txqueuelen 1000 (Ethernet)
    RX packets 442 bytes 61728 (61.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 491 bytes 50416 (50.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

schasse@TuxServeur:~$ sudo ifconfig enp0s3 up
schasse@TuxServeur:~$ ifconfig enp0s3
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::2435:4765:74d0:107a prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:97:14:04 txqueuelen 1000 (Ethernet)
    RX packets 455 bytes 63672 (63.6 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 518 bytes 53801 (53.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

On remarque qu'après un "down", l'interface n'a plus d'adresse IP car la ligne "inet" n'est plus là. On remarque aussi que le mot "UP" n'est pas présent non plus.

Par contre, en tapant à nouveau la commande avec up, l'adresse IP est réassociée.

La commande route

La commande **route** a déjà été entrevue un peu plus haut, avec la commande **ifconfig**. Le routage définit le chemin emprunté par les paquets entre son point de départ et son point d'arrivée. Cette commande permet également la configuration de pc, de switchs de routeurs.

Il existe 2 types de routages :

- le routage statique
- le routage dynamique.

Le routage statique consiste à imposer aux paquets la route à suivre.

Le routage dynamique met en oeuvre des algorithmes, qui permettent aux routeurs d'ajuster les tables de routage en fonction de leur connaissance de la topologie du réseau. Cette actualisation est réalisée par la réception des messages reçus des noeuds (routeurs) adjacents.

Le routage dynamique permet d'avoir des routes toujours optimisées, en fonction de l'état du réseau (nouveaux routeurs, engorgements, pannes).

On combine en général le routage statique sur les réseaux locaux au routage dynamique sur les réseaux

importants ou étendus.

Un administrateur qui dispose par exemple de 2 routeurs sur un réseau, peut équilibrer la charge en répartissant une partie du flux sur un port avec une route, et une autre partie sur le deuxième routeur.

Exemple de table de routage :

```
Kernel IP routing table
Destination Gateway      Genmask          Flags  Metric  Ref  Use  Iface
192.168.1.0   *          255.255.255.0   U      0        0    2    eth0
127.0.0.0     *          255.0.0.0       U      0        0    2    lo
default 192.168.1.9 0.0.0.0         UG     0        0   10    eth0
```

Commentaire généraux :

Destination : adresse de destination de la route

Gateway : adresse IP de la passerelle pour atteindre la route, * sinon

Genmask : masque à utiliser.

Flags : indicateur d'état (U - Up, H - Host - G - Gateway, D - Dynamic, M - Modified)

Metric : coût métrique de la route (0 par défaut)

Ref : nombre de routes qui dépendent de celle-ci

Use : nombre d'utilisation dans la table de routage

Iface : interface eth0, eth1, lo

Commentaire sur la 3ème ligne :

Cette ligne signifie que pour atteindre tous les réseaux inconnus, la route par défaut porte l'adresse 192.168.1.9. C'est la passerelle par défaut, d'où le sigle UG, G pour gateway.

Ajout d'une route

```
route add [net | host] addr [gw passerelle] [metric coût] [ netmask  
masque] [dev interface]
```

- net *ou* host indique l'adresse de réseau ou de l'hôte pour lequel on établit une route,

addr - adresse de destination,

gw - adresse de la passerelle,

metric - valeur métrique de la route,

netmask - masque de la route à ajouter,

dev - interface réseau à qui on associe la route.

Exemples :

```
route add 127.0.0.1 lo /* ajoute une route pour l'adresse 127.0.0.1 sur l'interface lo */
```

```
route add -net 192.168.2.0 eth0 /* ajoute une route pour le réseau 192.168.2.0 sur l'interface  
eth0 */
```

```
route add saturne.foo.org /* ajoute une route pour la machine machin sur l'interface eth0 */
```

```
route add default gw ariane /* ajoute ariane comme route par défaut pour la machine locale */
```

```
/* ariane est le nom d'hôte d'un routeur ou d'une passerelle */
```

```
/* gw est un mot réservé */
```

```
route add duschmoll netmask 255.255.255.192
```

```
/* Encore un qui a créé des sous réseaux., Il s'agit ici d'une classe C */
```

```
/* avec 2 sous réseaux, il faut indiquer le masque. */
```

Suppression d'une route

```
route del -net 192.168.1.0
```

```
route del -net monreseau.net
```

Attention

Attention, si on utilise des noms de réseau ou des noms d'hôtes, il faut qu'à ces noms soient associés les adresses de réseau ou des adresses IP dans le fichier `/etc/networks` pour les réseaux, et `/etc/hosts` ou DNS pour les noms d'hôtes.

La commande netstat

La commande **netstat**, permet de tester la configuration du réseau, visualiser l'état des connexions, établir des statistiques, notamment pour surveiller les serveurs.

Liste des paramètres utilisables avec **netstat** :

Sans argument, donne l'état des connexions,

- **a** afficher toutes les informations sur l'état des connexions,
- **i** affichage des statistiques,
- **C** rafraîchissement périodique de l'état du réseau,
- **n** affichage des informations en mode numérique sur l'état des connexions,
- **r** affichage des tables de routage,
- **t** informations sur les sockets TCP
- **u** informations sur les sockets UDP.

Etat des connexions réseau avec **netstat**, dont voici un exemple :

```
Proto Recv-Q Send-Q Local Address Foreign Address State
Tcp    0      126   uranus.planete.n:telnet 192.168.1.2:1037 ESTABLISHED
Udp    0        0   uranus.planet:netbios-dgm  *:*
Udp    0        0   uranus.plane:netbios-ns  *:*
```

Active UNIX domain sockets (w/o servers)

```
Proto RefCnt Flags      Type       State      I-Node Path
unix   2      [ ]       STREAM    Connected /dev/log
unix   2      [ ]       STREAM    Connected 1989
unix   1      [ ]       DGRAM          1955
```

Explications sur la première partie qui affiche l'état des connexions :

Proto : Protocole utilisé

Recv-q : nbre de bits en réception pour ce socket

Send-q : nbre de bits envoyés

LocalAddress : nom d'hôte local et port

ForeignAddress : nom d'hôte distant et port

State : état de la connexion

Le champ state peut prendre les valeurs suivantes:

Established : connexion établie

Syn snet : le socket essaie de se connecter

Syn rcv : le socket a été fermé

Fin wait2 : la connexion a été fermée

Closed : le socket n'est pas utilisé

Close wait : l'hôte distant a fermé la connexion; Fermeture locale en attente.

Last ack : attente de confirmation de la fermeture de la connexion distante

Listen : écoute en attendant une connexion externe.

Unknown : état du socket inconnu

Explications sur la deuxième partie qui affiche l'état des sockets (IPC - Inter Processus Communication) actifs :

Proto : Protocole, en général UNIX,

Refcnt : Nombre de processus associés au socket

Type : Mode d'accès datagramme (DGRAM), flux orienté connexion (STREAM), brut (RAW), livraison fiable des messages (RDM)

State : Free, Listening, Unconnected, connecting, disconnecting, unknown

Path : Chemin utilisé par les processus pour utiliser le socket.

Affichage et état des tables de routage avec netstat : netstat -nr ou netstat -r

Kernel IP routing table

Destination	Gateway	Genmask	Flags	MSS	Window	irtt	Iface
192.168.1.0	*	255.255.255.0	U	1500	0	0	eth0
127.0.0.0	*	255.0.0.0	U	3584	0	0	lo

Explications sur la commande netstat -r

Destination : adresse vers laquelle sont destinés les paquets

Gateway : passerelle utilisée, * sinon

Flags : G la route utilise une passerelle, U l'interface est active, H on ne peut joindre qu'un simple hôte par cette route)

Iface : interface sur laquelle est positionnée la route.

Affichage de statistiques avec netstat -i

Iface	MTU	Met	RX-OK	RX-ERR	RX-DRP	RX-OVR	TX-OK	TX-ERR	TX-DRP	TX-OVR	Flags
Lo	3584	0	89	0	0	0	89	0	0	0	BLRU
eth0	1500	0	215	0	0	0	210	0	0	0	BRU

Explications sur la commande netstat -i

RX-OK et TX-OK rendent compte du nombre de paquets reçus ou émis,

RX-ERR ou TX-ERR nombre de paquets reçus ou transmis avec erreur,

RX-DRP ou TX-DRP nombre de paquets éliminés,

RX-OVR ou TX-OVR recouvrement, donc perdus à cause d'un débit trop important.

Les Flags (B adresse de diffusion, L interface de loopback, M tous les paquets sont reçus, O arp est hors service, P connexion point à point, R interface en fonctionnement, U interface en service)

La commande traceroute

La commande **traceroute** permet d'afficher le chemin parcouru par un paquet pour arriver à destination. Cette commande est importante, car elle permet d'équilibrer la charge d'un réseau, en optimisant les routes.

Voici le résultat de la commande **traceroute www.nat.fr**, tapée depuis ma machine.

```
traceroute to sancy.nat.fr (212.208.83.2), 30 hops max, 40 byte packets
 1 195.5.203.9 (195.5.203.9) 1.363 ms 1.259 ms 1.270 ms
 2 194.79.184.33 (194.79.184.33) 25.078 ms 25.120 ms 25.085 ms
 3 194.79.128.21 (194.79.128.21) 88.915 ms 101.191 ms 88.571 ms
 4 cisco-eth0.frontal-gw.internext.fr (194.79.190.126) 124.796 ms[]
 5 sfinx-paris.remote-gw.internext.fr (194.79.190.250) 100.180 ms[]
 6 Internetway.gix-paris.ft.NET (194.68.129.236) 98.471 ms []
 7 513.HSSI0-513.BACK1.PAR1.inetway.NET (194.98.1.214) 137.196 ms[]
 8 602.HSSI6-602.BACK1.NAN1.inetway.NET (194.98.1.194) 101.129 ms[]
 9 FE6-0.BORD1.NAN1.inetway.NET (194.53.76.228) 105.110 ms []
10 194.98.81.21 (194.98.81.21) 175.933 ms 152.779 ms 128.618 ms[]
11 sancy.nat.fr (212.208.83.2) 211.387 ms 162.559 ms 151.385 ms[]
```

Explications :

Ligne 0 : le programme signale qu'il n'affichera que les 30 premiers sauts, et que la machine `www` du domaine `nat.fr`, porte le nom effectif de `sancy`, dans la base d'annuaire du DNS du domaine `nat.fr`. Cette machine porte l'adresse IP 212.208.83.2. Pour chaque tronçon, on a également le temps maximum, moyen et minimum de parcours du tronçon.

Ensuite, on a pour chaque ligne, l'adresse du routeur que le paquet a traversé pour passer sur le réseau suivant.

Ligne 4 et 5, le paquet a traversé 2 routeurs sur le même réseau 194.79.190.

Ligne 4, 5, 6, 7, 8, 9, 11, on voit que les routeurs ont un enregistrement de type A dans les serveurs de noms, puisqu'on voit les noms affichés.

Conclusion : depuis ma machine, chaque requête HTTP passe par 11 routeurs pour accéder au serveur `www.nat.fr`.

L'accès sur cet exemple est réalisé sur Internet. Un administrateur, responsable d'un réseau d'entreprise sur lequel il y a de nombreux routeurs, peut, avec cet outil, diagnostiquer les routes et temps de routage. Il peut ainsi optimiser les trajets et temps de réponse.

La commande dig

La commande **dig** remplace ce qui était la commande **nslookup**. Cette commande sert à diagnostiquer des dysfonctionnements dans la résolution de noms (Service DNS).

Utilisation simple de **dig** :

```
$ dig any freenix.org
; <<>> DiG 9.2.2 <<>> any freenix.org
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21163
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 3, ADDITIONAL: 3

;; QUESTION SECTION:
;freenix.org.                IN      ANY

;; ANSWER SECTION:
freenix.org.                92341  IN      SOA     ns2.freenix.org.\
                             hostmaster.freenix.org.\
                             2003042501\
                             21600\
                             7200\
                             3600000\
                             259200\

freenix.org.                117930 IN      NS      ns2.freenix.fr.
freenix.org.                117930 IN      NS      ns.frmug.org.
freenix.org.                117930 IN      NS      ns6.gandi.net.

;; AUTHORITY SECTION:
freenix.org.                117930 IN      NS      ns2.freenix.fr.
freenix.org.                117930 IN      NS      ns.frmug.org.
freenix.org.                117930 IN      NS      ns6.gandi.net.

;; ADDITIONAL SECTION:
ns2.freenix.fr.            16778  IN      A       194.117.194.82
ns.frmug.org.              40974  IN      A       193.56.58.113
ns6.gandi.net.             259119 IN      A       80.67.173.196

;; Query time: 197 msec
;; SERVER: 213.36.80.1#53(213.36.80.1)
;; WHEN: Tue May 27 15:16:23 2003
;; MSG SIZE rcvd: 248
```

retourne les informations sur le domaine concerné.

Il est ensuite possible d'interroger sur tout type d'enregistrement : SOA, MX, A, CNAME, PTR...