MODULE 3

Gestion du service réseau

Objectifs de ce module :

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

Table des matières

Sujets ***Page***

Introduction

Le service réseau est habituellement gérer 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.

Attention:

Le service NetworkManager est activé par défaut sur les systèmes de type RedHat, CentOS, etc. En fait, si votre installation est une installation de type ordinateur de bureau ou station de développement logiciel, alors le service NetworkManager sera activé à l'installation.

État du service NetworkManager

De façon à s'assurer que le service « NetworkManager » est bel et bien fonctionnel, nous allons faire afficher l'état de ce service. Pour y arriver, on peut exécuter la séquence suivante:

1. Vérifier que NetworkManager est présentement actif avec la commande systemctl:
   systemctl status NetworkManager

Vous devriez obtenir quelque chose qui ressemble à ceci :

```
NetworkManager.service - Network Manager
   Loaded: loaded (/usr/lib/systemd/system/NetworkManager.service; enabled)
   Active: active (running) since lun 2014-10-20 22:51:14 EDT; 1h 10min ago
 Main PID: 785 (NetworkManager)
   CGroup: /system.slice/NetworkManager.service
           ├─ 785 /usr/sbin/NetworkManager --no-daemon
           └─1354 /sbin/dhclient -d -sf /usr/libexec/nm-dhcp-helper -pf /var/run/dhclient-wlp3s0....
```

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.

On se connecte à un réseau en utilisant une carte ou une clef réseau. Du point de vue du système d'exploitation, ce périphérique est une *interface réseau*. Sous Linux, ces interfaces sont nommées *eth0, eth1…* pour des interfaces filaires, et *wlan0, wlan1* pour des interfaces sans fil (wifi, wimax…). Il existe également une interface spéciale, nommée *lo* (pour *loopback*) qui désigne toujours votre propre ordinateur.

> **Note**
>
> Nous supposerons que vous disposez d'une interface filaire, nommée *eth0*. Nous verrons plus loin comment afficher la liste de vos interfaces.

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.kubuntu.org*) et adresses IP (194.109.137.218).

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 pas 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

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  uranus.foo.org uranus
```

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/host.conf`

Il donne l'ordre dans lequel le processus de résolution de noms est effectué. Voici un exemple de ce que l'on peut trouver dans ce fichier :

```
order hosts,bind
```

La résolution est effectuée d'abord avec le fichier hosts, en cas d'échec avec le DNS.

Le fichier `/etc/resolv.conf`

Il permet d'affecter les serveurs de noms.

Exemple

```
Nameserver 192.168.1.1
Nameserver 192.168.1.2
Nameserver 192.168.1.3
```

Ici le fichier déclare le nom de domaine et les 3 machines chargées de la résolution de noms.

Les fichiers de configuration des interfaces réseau

Vous trouverez ces fichiers dans `/etc/sysconfig/network-scripts`. Chaque fichier dans ce répertoire contrôle un interface réseau. Lorsque le système démarre, celui-ci lit les fichiers s'y trouvant et détermine lequel des interfaces il doit démarrer et comment les configurer. Ces fichiers ont habituellement la forme suivante:

ifcfg-nom

où nom réfère au nom du périphérique que le fichier contrôle.

Les interfaces Ethernet

Le fichier le plus souvent rencontré est sans doute celui qui se nomme "ifcfg-eth0" qui contrôle la première carte réseau attachée au système. La deuxième carte réseau, s'il y en a une, se nomme alors "ifcfg-eth1" et ainsi de suite.

Voici un exemple de fichier relié à l'interface eth0. Ce fichier se nomme ifcfg-eth0 et configure l'interface pour configurer une adresse IP fixe :

```
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
NETMASK=255.255.255.0
IPADDR=192.168.1.100
USERCTL=no
```

Exemple 1: Assigner une adresse IP statique (192.168.1.150 par exemple) à eth0

On édite le fichier ifcfg-eth0 (ou on le créé au besoin s'il n'est pas encore présent) et on y ajoute:

```
DEVICE=eth0
#c'est ici que l'on mentionne que l'adresse est statique
BOOTPROTO=static
#L'interface sera active au démarrage de l'ordinateur
ONBOOT=yes
NETMASK=255.255.255.0
IPADDR=192.168.1.100
USERCTL=no
```

Exemple 2: Assigner une adresse IP par DHCP à eth0

On édite le fichier (ou on le créé au besoin s'il n'est pas encore présent) et on y ajoute:

```
DEVICE=eth0
#c'est ici que l'on mentionne que l'adresse est attribuée par DHCP
BOOTPROTO=dhcp
#L'interface sera active au démarrage de l'ordinateur
ONBOOT=yes
#Les usagers qui ne sont pas“root“ ne peuvent pas contrôler cet interface.
USERCTL=no
```

Il s'agira de repartir le service réseau de la façon suivante :

systemctl restart NetworkManager

pour que les modifications soient prises en compte.

Changement de nom de périphérique réseau

Il est possible de changer le nom que Fedora utilise pour les cartes réseaux de votre système, En effet, à partir de Fedora 20, les cartes réseaux cablées obtiennent des noms qui commencent par « em ». Les cartes sans fil obtiennent des noms qui débutent par « wlp... ».

Historiquement, les cartes réseaux ont toujours eux des noms qui débutaient par « eth » suivi d'un numéro qui identifiait la carte des autres cartes dans le système. Ainsi la première carte portait le nom « eth0 », la deuxième carte « eth1 ». Nous allons procéder à des fins de démonstration qu'il est possible de revenir à un schéma de style « eth ».

Voici la séquence à suivre :

- Se déplacer dans le répertoire « /etc/sysconfig/network-script ».
- Éditer le fichier nommé « ifcfg-em1 » qui correspond à la première carte ethernet physique.

- Dans ce fichier, ajoutez :
  DEVICE=eth0

  et mettez en commentaire la ligne qui débute par « Name= ».

- Sauvegardez le fichier sous le nom « ifcfg-eth0 ».
- Repartez l'ordinateur : sudo reboot

Lorsque votre ordinateur est reparti, exécutez la commande « ifconfig » et constatez que vous avez maintenant votre carte réseau filaire qui se nomme « eth0 ».

Alias d'adresse IP

Il est possible de « simuler » une autre carte réseau de façon indépendante de la carte physique réelle.

Pour y arriver :

- Allez dans le répertoire « /etc/sysconfig/network-script »
- Créez un nouveau fichier et entrez les informations suivantes :

```
TYPE=Ethernet
DEVICE=eth0:1
BOOTPROTO=static
DEFROUTE=yes
ONBOOT=yes
#HWADDR=  ATTENTION ENTREZ ICI l'adresse MAC de la carte
#réelle que vous voulez dupliquer
IPADDR=192.168.1.100
NETMASK=255.255.255.0
```

Sauvegardez le fichier sous le nom « ifcfg-eth0:1 » et exécutez la commande :
sudo ifup eth0:1

Adresse IP et masque de sous-réseau

La configuration IP proprement dite peut être gérée à l'aide de la commande **ifconfig**. Sans argument, celle-ci affiche la configuration de toutes vos interfaces réseau configurées :

```
% ifconfig
eth0      Link encap:Ethernet  HWaddr 00:1c:23:3f:ff:bb
          inet adr:192.168.0.105  Bcast:192.168.0.255  Masque:255.255.255.0
          adr inet6: fe80::21c:23ff:fe3f:ffbb/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:749880 errors:0 dropped:0 overruns:0 frame:0
          TX packets:393902 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:910931782 (868.7 MiB)  TX bytes:32422248 (30.9 MiB)
          Interruption:17

lo        Link encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128 Scope:Hôte
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:71 errors:0 dropped:0 overruns:0 frame:0
          TX packets:71 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          RX bytes:10096 (9.8 KiB)  TX bytes:10096 (9.8 KiB)
```

> **Note**
>
> Cette commande peut également afficher les interfaces non configurées, avec l'argument **-a**, pour *all*. Cela peut servir à déterminer le nom d'une interface non encore configurée.

Pour appliquer une configuration à une interface, on utilise la même commande, avec des arguments correspondants à cette configuration. Par exemple, pour utiliser l'adresse IPv4 192.168.0.42 et le masque de sous-réseau 255.255.255.0 sur l'interface *eth0*, en tant que root, tapez :

```
ifconfig eth0 192.168.0.42
```

Pour ajouter l'adresse IPv6 2001:db8::6726 avec un sous-réseau de longueur 32, tapez :

```
ifconfig eth0 add 2001:db8::6726/32
```

> **Note**
>
> Pour supprimer une configuration IPv4, utilisez la commande **ifconfig eth0 down**. Pour supprimer une configuration IPv6, utilisez la commande **ifconfig eth0 del <adresse>**.

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 conmplète, 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 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 `eth0 192.168.1.2` (affecte l'adresse 192.168.1.2 à la première interface physique).

Voici les principaux arguments utilisés :

Nom du paramètres	Description
`eth0, eth1, wlan0, wlan1...`	`l'interface` logique ou physique, il est obligatoire,
`Up`	Active l'interface
down	Désactive l'interface
`mtu`	définit l'unité de transfert des paquets
`netmask`	affecter un masque de sous-réseau
`broadcast`	définit l'adresse de broadcast
`arp` ou `-arp`	active ou désactive l'utilisation du cache arp de l'interface
`metric`	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 *promiscuous*, 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.

Description du résultat de la commande **ifconfig** `eth0` *:*

1. eth0 Link encap:Ethernet HWaddr 00:80:C8:32:C8:1E
2. inet addr:192.168.1.1 Bcast:192.168.1.255 Mask:255.255.255.0
3. UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
4. RX packets:864 errors:0 dropped:0 overruns:0 frame:0
5. TX packets:654 errors:0 dropped:0 overruns:0 carrier:0
6. collisions:0
7. Interrupt:10 Base address:0x6100

Explications :

Ligne 1: l'interface est de type Ethernet. La commande nous donne l'adresse MAC de l'interface.

Ligne 2 : on a l'adresse IP celle de broadcast, celle du masque de sous-réseau

Ligne 3 : l'interface est active (UP), les modes broadcast et multicast le sont également, le MTU est de 1500 octets, le Metric de 1

Ligne 4 et 5 : RX (paquets reçus), TX (transmis), erreurs, suppressions, engorgements, collision

Mode d'utilisation :

Ce paragraphe décrit une suite de manipulation de la commande **ifconfig**.

Ouvrez une session en mode console sur une machine.

1 - Relevez les paramètres de votre machine à l'aide de la commande **ifconfig**. Si votre machine n'a qu'une interface physique, vous devriez avoir quelque chose d'équivalent à cela.

```
Lo  Link encap:Local Loopback
inet addr:127.0.0.1  Bcast:127.255.255.255  Mask:255.0.0.0
UP BROADCAST LOOPBACK RUNNING  MTU:3584  Metric:1
RX packets:146 errors:0 dropped:0 overruns:0 frame:0
TX packets:146 errors:0 dropped:0 overruns:0 carrier:0
collisions:0

eth0 Link encap:Ethernet  HWaddr 00:80:C8:32:C8:1E
inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:864 errors:0 dropped:0 overruns:0 frame:0
TX packets:654 errors:0 dropped:0 overruns:0 carrier:0
collisions:0

Interrupt:10 Base address:0x6100
```

2 - Désactivez les 2 interfaces lo et eth0

ifconfig *lo* **down**

ifconfig *eth0* down

3 - Tapez les commandes suivantes :

ping *localhost*

ping *192.168.1.1*

telnet *localhost*

Aucune commande ne fonctionne, car même si la configuration IP est correcte, les interfaces sont désactivées.

4 - Activez l'interface de loopback et tapez les commandes suivantes :

`ifconfig` *lo* **up** /* activation de l'interface de loopback */

`ping localhost` ou telnet localhost /* ça ne marche toujours pas */

`route add` *127.0.0.1* /* on ajoute une route sur l'interface de loopback */

ping *localhost* ou `telnet` *localhost* /* maintenant ça marche */

`ping` *192.168.1.1* /* ça ne marche pas car il manque encore une route*/

On peut déduire que :

- pour chaque interface il faudra indiquer une route au protocole.
- dans la configuration actuelle, aucun paquet ne va jusqu'à la carte, donc ne sort sur le réseau.

Voici le rôle de l'interface loopback. Elle permet de tester un programme utilisant le protocole IP sans envoyer de paquets sur le réseau. Si vous voulez écrire une application réseau, (telnet, ftp, ou autre), vous pouvez la tester de cette façon.

5 - Activez l'interface eth0 et tapez les commandes suivantes :

`ifconfig eth0` **up** /* activation de l'interface */

`route` **add** `192.168.1.1`

ifconfig /* l'information Tx/Rx de l'interface eth0 vaut 0 */

/* Aucun paquet n'est encore passé par la carte.*/

`ping` *127.0.0.1*

`ifconfig` /* on voit que l'information Tx/Rx de lo est modifiée */

/* pas celle de eth0, on en déduit que les paquets */

/* à destination de lo ne descendent pas jusqu'à l'interface physique */

`ping` *192.168.1.1* /* test d'une adresse locale */

`ifconfig` /* Ici on peut faire la même remarque. Les paquets ICMP */

/* sur une interface locale, ne sortent pas sur le réseau */

/* mais ceux de l'interface lo sont modifiés*/

ping 192.168.1.2 /* test d'une adresse distante */

`ifconfig` /* Ici les paquets sont bien sortis. Les registres TX/RX de eth0 */

/* sont modifiés, mais pas ceux de lo */

6 -Réalisez les manipulations suivantes, nous allons voir le comportement de la commande **ping** sur les interfaces.

Sur la machine 192.168.1.1, tapez la commande

ifconfig /* relevez les valeurs des registres TX/RX */

sur la machine ayant l'adresse 192.168.1.2, tapez:

ping 192.168.1.1

Sur la machine 192.168.1.1, tapez

ifconfig /* relevez les nouvelles valeurs des registres TX/RX */

/* il y a bien eu échange Réception et envoi de paquets*/

192.168.1.2 ping 192.168.1.3

192.168.1.1 ifconfig /* On voit que le registre Rx est modifié mais */

/* le registre Tx n'est pas modifié. La machine a bien reçu*/

/* le paquet mais n'a rien renvoyé */

192.168.1.2 ping 192.168.1.2

192.168.1.2 ifconfig /* aucun registre n'est modifié, donc les paquets */

/* ne circulent pas jusqu'à l'interface physique avec un ping*/

/* sur l'interface locale */

7 - le MTU (*Message Transfert Unit*) détermine l'unité de transfert des paquets.

Vous allez, sur la machine 192.168.1.1 modifier le MTU par défaut à 1500, pour le mettre à 300, avec la commande :

`ifconfig` *eth0 mtu 300*

Sur la machine d'adresse 192.168.1.2, vous allez ouvrir une session ftp et chronométrer le temps de transfert d'un fichier de 30 MO. Relevez le temps et le nombre de paquets transmis ou reçus (commande **ifconfig**, flags TX/RX).

Restaurez le paramètre par défaut sur la première machine.

Refaites le même transfert et comparez les chiffres. La différence n'est pas énorme sur le temps car le volume de données est peu important. Par contre la différence sur le nombre de paquets, elle, est importante.

La commande arp

Description de la commande

La commande **arp** permet de visualiser ou modifier la table du cache arp de l'interface. Cette table peut être statique et (ou) dynamique. Elle donne la correspondance entre une adresse IP et une adresse **MAC** (Ethernet).

A chaque nouvelle requête, le cache ARP de l'interface est mis à jour. Il y a un nouvel enregistrement. Cet enregistrement à une durée de vie (ttl ou *Time To Live*).

Voici un exemple de cache ARP obtenu avec la commande **arp** `-va` :

```
? (192.168.1.2) at 00:40:33:2D:B5:DD [ether] on eth0
>Entries: 1      Skipped: 0      Found: 1
```

On voit l'adresse IP et l'adresse MAC correspondante. Il n'y a qu'une entrée dans la table. Voici les principales options de la commande **arp** :

arp `-s` (ajouter une entrée statique), exemple : **arp** `-s 192.168.1.2 00:40:33:2D:B5:DD`

arp `-d` (supprimer une entrée), exemple : **arp** `-d 192.168.1.2`

Voir la page **man** pour les autres options.

La table ARP et le fonctionnement du cache ARP.

Cela est réalisé par la configuration de tables ARP statiques.

Mode d'utilisation :

Attention à certaines interprétations de ce paragraphe. Il dépend de votre configuration. Soit vous êtes en réseau local avec une plage d'adresse déclarée, soit vous utilisez une carte d'accès distant.

Première partie :

1. Affichez le contenu de la table ARP avec la commande **arp -a**,
2. Supprimez chaque ligne avec la commande **arp -d** @*ip*, où `@ip` est l'adresse IP de chaque hôte apparaissant dans la table,
3. La commande **arp** `-a` ne devrait plus afficher de ligne,
4. Faites un **ping**, sur une station du réseau local,
5. **arp** `-a`, affiche la nouvelle entrée de la table,
6. Ouvrez une session sur Internet, puis ouvrez une session ftp anonyme sur un serveur distant en utilisant le nom, par exemple `ftp.cdrom.com`. Utilisez une adresse que vous n'avez jamais utilisée, supprimez également tout gestionnaire de cache.
7. Affichez le nouveau contenu de la table avec **arp** `-a`. Le cache ARP ne contient pas l'adresse Ethernet du site distant, mais celle de la passerelle par défaut. Cela signifie que le client n'a pas à connaître les adresses Ethernet des hôtes étrangers au réseau local, mais uniquement l'adresse de la passerelle. Les paquets sont ensuite pris en charge par les routeurs.

8. Refaites une tentative sur le site choisi précédemment. Le temps d'ouverture de session est normalement plus court. Cela est justifié, car les serveurs de noms ont maintenant dans leur cache la correspondance entre le nom et l'adresse IP.

Deuxième partie :

La commande **arp** permet de diagnostiquer un dysfonctionnement quand une machine prend l'adresse IP d'une autre machine.

1. Sur la machine 192.168.1.1, faites un **ping** sur 2 hôtes du réseau 192.168.1.2 et 192.168.1.3,
2. A l'aide de la commande **arp**, relevez les adresses MAC de ces noeuds,
3. Modifiez l'adresse IP de la machine 192.168.1.2 en 192.168.1.3
4. relancez les 2 machines en vous arrangeant pour que la machine dont vous avez modifié l'adresse ait redémarré la première,
5. Sur la machine d'adresse 192.168.1.1, remettez à jour les tables ARP.
6. Quel est le contenu, après cela de la table ARP ?

Conclusion : vous allez avoir un conflit d'adresses. Vous allez pouvoir le détecter avec la commande **arp**. Autre problème, si vous faites un **telnet** sur 192.168.1.3, il y a de fortes chances pour que ce soit la machine qui était d'adresse 192.168.1.2, qui vous ouvre la session. Nous sommes arrivés à mettre la pagaille sur un réseau de 3 postes. Cette pagaille pourrait tourner vite au chaos sur un grand réseau, d'où la nécessité pour un administrateur de faire preuve d'une grande rigueur.

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 un 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.plan: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      1990     /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

LocalAdress : nom d'hôte local et port

ForeignAdress : 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 recv : 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'intérroger sur tout type d'enregistrement : SOA, MX, A, CNAME, PTR...

La commande host

La commande **host** interroge les serveurs de noms. Elle peut par exemple être utilisée pour détecter des dysfonctionnement sur un réseau (serveurs hors services). Attention, n'utilisez pas cette commande sur des réseaux dont vous n'avez pas l'administration.