## ifconfig : liste des interfaces réseau

Nous allons découvrir ici deux commandes : `ifconfig` et `netstat`. La première permet de gérer les connexions réseau de votre machine (pour les activer / désactiver, par exemple) tandis que la seconde vous permet d'analyser ces connexions, de connaître des statistiques, etc.

Votre ordinateur possède en général plusieurs **interfaces réseau**, c'est-à-dire plusieurs moyens de se connecter au réseau.

Tapez `ifconfig` dans la console pour voir ce que ça donne :

```
$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:90:f5:56:44:5a
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          Packets reçus:0 erreurs:0 :0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:0 (0.0 B) Octets transmis:0 (0.0 B)
          Interruption:220 Adresse de base:0xe000

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
          Packets reçus:10 erreurs:0 :0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          Octets reçus:500 (500.0 B) Octets transmis:500 (500.0 B)

wlan0     Link encap:Ethernet  HWaddr 00:19:d2:61:90:0a
          inet adr:192.168.1.2  Bcast:192.168.1.255  Masque:255.255.255.0
          adr inet6: fe80::219:d2ff:fe61:900a/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Packets reçus:5238 erreurs:0 :0 overruns:0 frame:0
          TX packets:4899 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:5069449 (5.0 MB) Octets transmis:1202459 (1.2 MB)
```

On distingue ici trois interfaces réseau. Vous en avez peut-être plus, peut-être moins ; tout dépend de votre ordinateur.

Les interfaces que j'ai sont assez courantes, détaillons-les :

- `eth0` : cela correspond à la connexion par câble réseau (ce qu'on appelle en général le *câble RJ45* – figure suivante). Si votre PC est relié au réseau via un câble, c'est sûrement ce moyen de communication que vous utilisez actuellement. Notez que certains ordinateurs (et notamment les serveurs) ont plusieurs sorties réseau filaires. Dans ce cas, vous devriez voir aussi des interfaces `eth1`, `eth2`, etc.
- `lo` : c'est la boucle locale. Tout le monde devrait avoir cette interface. Elle correspond à une connexion à… vous-mêmes. C'est pour cela qu'on l'appelle la boucle locale : tout ce qui est envoyé par là vous revient automatiquement. Cela peut paraître inutile, mais on a parfois besoin de se connecter à soi-même pour des raisons pratiques.

- `wlan0` : il s'agit d'une connexion sans-fil type Wi-Fi. Là encore, bien que ce soit plus rare, si vous avez plusieurs cartes réseau sans fil, vous aurez un `wlan1`, `wlan2`, etc.

Observez les résultats de ma commande et essayez de deviner par quelle interface réseau je me connecte à l'internet.

…

Vous avez trouvé ? Il ne fallait pas avoir peur de lire le détail des messages.
En effet, bien que je possède une sortie réseau filaire (RJ45), j'utilise ici le Wi-Fi, comme en témoigne la ligne `Packets reçus:5238` pour le Wi-Fi `wlan0` (alors qu'il y en a 0 pour `eth0`). C'est donc l'interface active que j'utilise le plus.

La commande `ifconfig` permet aussi de faire des réglages réseau. Toutefois, cela sortirait un peu du cadre de ce cours et il vous faudrait des connaissances en réseau pour bien l'utiliser.
Voici cependant un réglage très simple que vous pouvez faire et qui vous sera probablement utile : l'activation / désactivation d'interface.

Il suffit d'écrire une commande sous cette forme :

```
ifconfig interface etat
```

Remplacez :

- `interface` par le nom de l'interface que vous voulez modifier (`eth0`, `wlan0`…) ;
- `etat` par `up` ou `down` selon si vous voulez activer ou désactiver l'interface.

Exemple :

```
$ ifconfig eth0 down
```

… désactive l'interface `eth0` (filaire). Plus aucun trafic ne pourra alors circuler par l'interface `eth0`.

```
$ ifconfig eth0 up
```

… la réactive de nouveau.

Vous aurez peut-être besoin de connaître ces commandes un jour ou l'autre si vous devez désactiver puis réactiver une interface pour prendre en compte des changements dans la configuration de votre réseau.

### `netstat` : statistiques sur le réseau

La commande `netstat` risque de vous paraître un peu complexe si vous avez peu de connaissances concernant les réseaux, mais elle est incontournable quand on veut savoir ce que notre machine est en train de faire sur le réseau.

`netstat` peut afficher beaucoup d'informations. Pour sélectionner celles qui nous intéressent, on a recours à de nombreux paramètres.
Plutôt que de les expliquer un par un, je vais vous montrer quelques combinaisons de paramètres qui donnent des résultats intéressants.

#### `netstat -i` : statistiques des interfaces réseau

Pour commencer, essayez l'option `-i` :

```
$ netstat -i
Table d'interfaces noyau
Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0       1500 0          0      0      0 0             0      0      0      0 BMU
lo        16436 0         10      0      0 0            10      0      0      0 LRU
wlan0      1500 0       5161      0      0 0          4810      0      0      0 BMRU
```

Vous n'aurez pas nécessairement les mêmes lignes que moi ; tout dépend de votre ordinateur.
Il s'agit là d'un tableau présentant, pour chaque interface réseau que vous avez, une série de statistiques d'utilisation. On retrouve ici nos interfaces `eth0`, `lo` et `wlan0`.

Comme vous le voyez sur la colonne `RX-ERR`, c'est `wlan0` qui est l'interface la plus active. Et vous noterez que `lo` est un petit peu utilisée elle aussi ; comme quoi se connecter à soi-même peut s'avérer utile.

Je ne rentrerai pas dans le détail de ces colonnes car c'est assez technique, mais vous savez au moins détecter l'activité de vos interfaces grâce à cette commande.

**`netstat -uta` : lister toutes les connexions ouvertes**

```
$ netstat -uta
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale          Adresse distante        Etat
tcp        0      0 *:ssh                   *:*                     LISTEN
tcp        0      0 localhost:ipp           *:*                     LISTEN
tcp        0      0 mateo21-laptop.lo:60997 debian-mirror.mirro:ftp ESTABLISHE
tcp        1      0 mateo21-laptop.lo:33721 lisa.simple-it.fr:www CLOSE_WAIT
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
udp        0      0 *:bootpc                *:*
udp        0      0 *:mdns                  *:*
udp        0      0 *:45176                 *:*
```

Les options signifient :

- `-u` : afficher les connexions UDP ;
- `-t` : afficher les connexions TCP ;
- `-a` : afficher toutes les connexions quel que soit leur état.

TCP et UDP sont deux protocoles différents pour envoyer des données sur le réseau.
UDP est plutôt utilisé dans les jeux en réseau et pour les communications vocales (avec Skype, par exemple). Sinon, de manière générale, TCP est le protocole le plus utilisé. Je n'irai pas plus loin dans les explications mais vous pouvez vous renseigner si le sujet vous intéresse.

Pour filtrer un peu, on va enlever les connexions UDP qui, la plupart du temps, sont moins importantes :

```
$ netstat -ta
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale          Adresse distante        Etat
tcp        0      0 *:ssh                   *:*                     LISTEN
tcp        0      0 localhost:ipp           *:*                     LISTEN
tcp        0      0 mateo21-laptop.lo:60997 debian-mirror.mirro:ftp ESTABLISHE
tcp        0   4107 mateo21-laptop.lo:33721 lisa.simple-it.fr:www ESTABLISHED
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
```

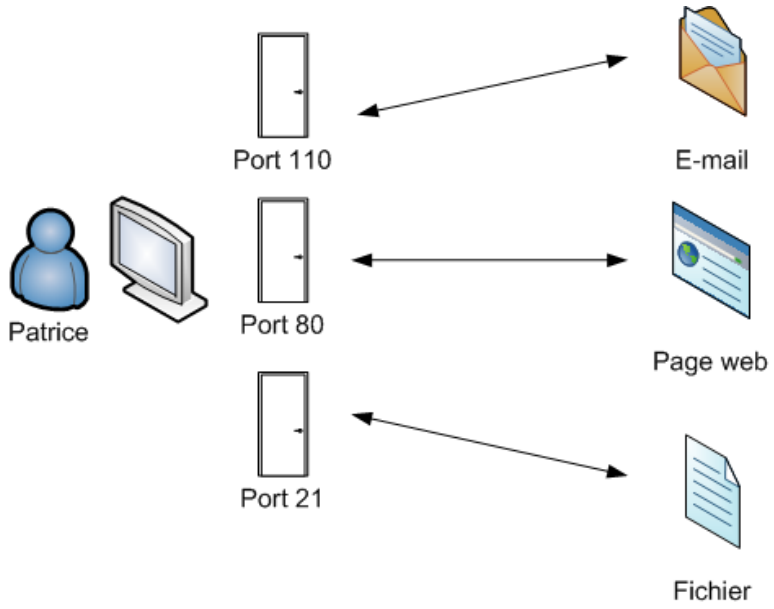
Ce tableau vous indique qui, depuis l'adresse locale, est connecté à qui (à une adresse distante).

Chaque connexion a un état. Ici, on repère les états `LISTEN` et `ESTABLISHED`.
De nombreux états sont possibles ; en voici quelques-uns à connaître :

- `ESTABLISHED` : la connexion a été établie avec l'ordinateur distant ;
- `TIME_WAIT` : la connexion attend le traitement de tous les paquets encore sur le réseau avant de commencer la fermeture ;
- `CLOSE_WAIT` : le serveur distant a arrêté la connexion de lui-même (peut-être parce que vous êtes restés inactifs trop longtemps ?) ;
- `CLOSED` : la connexion n'est pas utilisée ;
- `CLOSING` : la fermeture de la connexion est entamée mais toutes les données n'ont pas encore été envoyées ;
- `LISTEN` : à l'écoute des connexions entrantes.

Il y en a d'autres que vous pouvez lire dans la documentation. Globalement, ce qu'il faut retenir, c'est que les connexions à l'état `LISTEN` ne sont pas utilisées actuellement mais qu'elles « écoutent » le réseau au cas où quelqu'un veuille se connecter à votre ordinateur.

Regardez en particulier le **port** sur lequel ces connexions écoutent (après le symbole « : ») car c'est probablement l'information la plus intéressante. En effet, on peut se connecter à chaque ordinateur via différentes « portes » appelées *ports*. Chaque service utilise un port différent, comme l'illustre la figure suivante.




À la première ligne, vous avez `*:ssh`, ce qui signifie que SSH est en train d'écouter sur le port de SSH au cas où quelqu'un veuille se connecter à votre machine. C'est logique puisque j'ai activé le serveur SSH pour pouvoir m'y connecter à distance au besoin.

D'autres connexions, elles, sont déjà établies et donc en cours d'utilisation. Par exemple, au niveau de l'adresse distante, je suis connecté par FTP à `debian-mirror.mirro:ftp` et je suis connecté à un serveur web `lisa.simple-it.fr:www`.
En clair, je suis en train de charger une page sur le Site du Zéro.

Vous pouvez ajouter `-n` si vous désirez avoir les numéros des ports plutôt qu'une description en toutes lettres :

```
$ netstat -tan
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale          Adresse distante        Etat
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp       15      0 192.168.1.2:60997       128.101.240.212:21      CLOSE_WAIT
tcp        0      0 192.168.1.2:54001       80.248.219.123:80       ESTABLISHE
tcp6       0      0 :::22                   :::*                    LISTEN
```

Cela correspond aux ports que l'on connaît : 22 pour SSH, 21 pour FTP, 80 pour le web, etc.

**`netstat -lt` : liste des connexions en état d'écoute**

Très utile, l'option `-l` vous permet de filtrer les connexions à l'état `LISTEN` et donc de savoir quels ports de serveur sont susceptibles d'être utilisés en ce moment sur votre machine.

```
$ netstat -lt
Connexions Internet actives (seulement serveurs)
Proto Recv-Q Send-Q Adresse locale          Adresse distante        Etat
tcp        0      0 *:ssh                   *:*                     LISTEN
tcp        0      0 localhost:ipp           *:*                     LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
```

**`netstat -s` : statistiques résumées**

Enfin, si vous êtes très friands de statistiques réseau, `-s` est fait pour vous :

```
$ netstat -s
Ip:
    7443 paquets reçus au total
    1 avec des en-têtes invalides
    8 avec des adresses invalides
    0 réacheminés
    0 paquets arrivant rejetés
    7354 paquets entrants délivrés
    7226 requêtes envoyées
Icmp:
    0 Messages ICMP reçus
    0 messages ICMP entrant échoués

[...]
```