TP1 : Prise en main des commandes réseaux sous Unix

Rabii El ghorfi

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 :

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` activer ou désactiver 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 */

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.

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.

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 ou suppression d'une route :

route **add [net | host] addr [gw passerelle] [métric coût] [netmask masque] [dev interface]**

- *net* ou *host* indique l'adresse de réseau ou de l'hôte pour lequel on établit une route,
- adresse de destination,
- adresse de la passerelle,
- valeur métrique de la route,
- masque de la route à ajouter,
- 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 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** ***toutbet-net***

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.