La configuration réseau sous Linux

Ludovic Terrier et Arnaud Goulut

Mars 2010

1 Partie 1

1.1 Plan de câblage du réseau

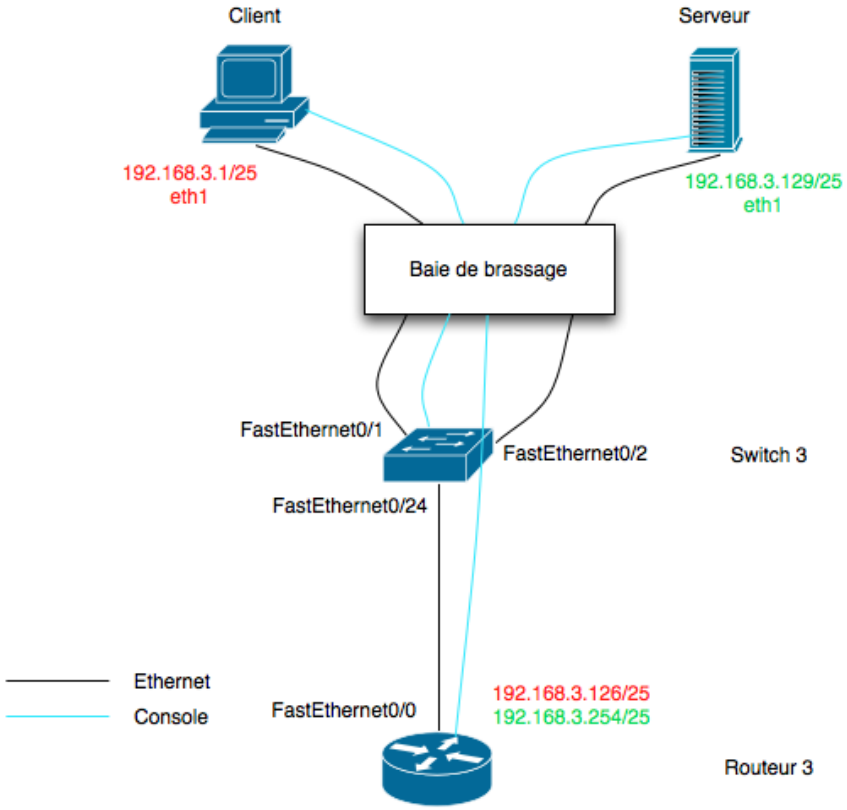

FIGURE 1 – Présentation du réseau configuré au cours du TP.

1.2 Configuration IP

Dans les réseaux du serveur et du client on utilise un seul réseau (192.168.3.0) de classe C. Cependant pour en faire deux réseaux distincts on le scinde à l'aide du masque : `/25`. Ceci signifie que les 25 premiers bits de chaque adresse IP utilisée sur le réseau correspondra au NetID et les 7 derniers au HostID. On obtient donc 2 sous-réseaux ayant pour adresse respective : 192.168.3.0 et 192.168.3.128.

Dans le cas du réseau de transport, nous avons un réseau 172.16.3.0, qui contiendra seulement deux machines. C'est pourquoi on peut se permettre d'utiliser un masque `/30` qui, appliqué à notre réseau de départ, donne 4 adresses :

- deux adresses à utiliser pour des hosts,
- une pour le broadcast,
- une pour le NetID.

Ci-dessous les configurations des 3 réseaux :

Réseau du client	
NetID	192.168.3.0
Masque	/25
Plage d'adresse	192.168.3.0 - 192.168.3.127
1ere adresse machine	192.168.3.1
Dernière adresse machine	192.168.2.126
Nombre de machines potentiel	126
Broadcast	192.168.3.127

Réseau du seveur (DMZ)	
NetID	192.168.3.128
Masque	/25
Plage d'adresse	192.168.3.128 - 192.168.3.255
1ere adresse machine	192.168.3.129
Dernière adresse machine	192.168.3.254
Nombre de machines potentiel	126
Broadcast	192.168.3.255

Réseau de transport	
NetID	172.16.3.0
Masque	/30
Plage d'adresse	172.16.3.0 - 172.16.3.3
1ere adresse machine	172.16.3.1
Dernière adresse machine	172.16.3.2
Nombre de machines potentiel	2
Broadcast	172.16.3.3

1.3 Configuration des équipements via le port console

Pour l'administration de matériel réseaux on préfère utiliser un port console dédié. Ceci afin d'assurer l'accès à l'équipement en cas de mauvaise manipulation, qui le rendrait inaccessible via ses interfaces réseaux (Ethernet dans le cadre des TPs) ou dans le cas d'une congestion du réseau. On s'affranchit alors de tout problème d'adressage.

Cette interface a un port série (avec connectique RJ45 et DB9)

2 Partie 2

2.1 Configuration du switch

Deux sous-réseaux sont connectés au switch, c'est pourquoi nous devons le séparer logiquement. Ceci est effectué grâce aux Virtual Local Area Networks (VLANs). Une différenciation des deux VLANs peut se faire par ports, ce qui se voit dans le fichier de configuration des switchs, seul le port *FastEthernet0/2* est attribué à un VLAN :

```
⋮
interface FastEthernet0/2
switchport access vlan 2
⋮
```

Ce qui peut vouloir dire que l'ensemble des autres ports sont dans le même VLAN par défaut.

2.2 Configuration du routeur

Le switch de notre réseau étant relié au routeur par un seul médium et que celui-ci transporte les flux de deux sous-réseaux, il faut que l'interface routeur soit elle-même séparée en deux interfaces logiques. Ce qui semble fait au regard du fichier de configuration du routeur :

```
        ⋮
interface FastEthernet0/0.1
encapsulation dot1Q 1 native
ip address 192.168.3.126 255.255.255.128
!
interface FastEthernet0/0.2
encapsulation dot1Q 2
ip address 192.168.3.254 255.255.255.128
        ⋮
```

L'interface *FastEthernet0/0* du routeur à laquelle est connectée le switch est partagée en deux :

- **.1** pour le réseau 192.168.3.0 (l'interface ayant l'adresse 192.168.3.126)
- **.2** pour le réseau 192.168.3.128 (l'interface ayant l'adresse 192.168.3.254)

3 Partie 3

3.1 /etc/sysconfig/network

Pour communiquer sur un réseau, une machine nécessite une passerelle par défaut, mais peut aussi avoir besoin d'un nom. Ce sont donc ces informations que l'on s'autorise à stocker dans le fichier `/etc/sysconfig/network`

```
HOSTNAME= azerty
GATEWAY=192.168.1.254
```

3.2 Pilote de la carte réseau

En lisant le fichier `/etc/modprobe.conf` on peut lire la ligne :

```
        ⋮
alias eth1 e1000e
        ⋮
```

Ce qui signifie que l'alias *eth1* est créé vers le pilote (*e1000e*) de la carte réseau. Si l'on effectue la commande *ifconfig* dans un Terminal on pourra observer *eth1* et pas *e1000e*.

3.3 Paramètres de l'interface réseau

Les paramètres que l'on peut attribuer à une interface réseau avec la commande *ifconfig* sont l'adresse IP et le masque de sous réseau. Ceci est effectué grâce à la commande suivante :

```
ifconfig <interface> <adresse_IP> netmask <masque>
```

Pour le Client :

```
ifconfig eth1 192.168.3.1 netmask 255.255.255.128
```

Pour le Serveur :

```
ifconfig eth1 192.168.3.129 netmask 255.255.255.128
```

On peut aussi, à l'aide de cette commande, activer ou désactiver l'interface :

```
ifconfig eth1 up
```

```
ifconfig eth1 down
```

3.4 Pérenniser les paramètres de l'interface réseau

Les informations concernant l'adresse IP et le masque peuvent-être conservées de manière pérenne dans le fichier : /etc/sysconfig/network − scripts/ifcfg− < interface > ou dans notre cas /etc/sysconfig/network − scripts/ifcfg − eth1

Il suffit de l'éditer et d'y ajouter :

```
IPADDR=<adresse_IP>
NETMASK=<masque>
```

Pour le poste Client :

```
IPADDR=192.168.0.1
NETMASK=255.255.255.0
```

3.5 Table de routage

La table de routage donne à une machine le premier saut qui lui permet d'atteindre n'importe quelle autre machine située n'importe où sur l'Internet.

```
route add -net <reseau_destination> netmask <masque> <interface>
```

Où <interface> est l'interface à utiliser pour atteindre la destination.

De plus, cette table contient une route par défaut qui est utilisée par la machine en dernier recours. On parle alors de *passerelle* ou de *gateway*. Pour la signifier à une machine on tape la commande :

```
route add default gw <adresse_IP_passerelle>
```

Exemple d'une table de routage affichée par la commande : *route*.

```
[root@server][~] route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     *               255.255.255.128 U     0      0        0 eth0
default         192.168.1.126   0.0.0.0         UG    0      0        0 eth0
```

Considérons la machine 192.168.1.23 avec la table ci-dessus, désirant communiquer avec 192.168.1.34. La machine va donc lire sa table de routage, s'apercevoir que la destination appartient au même sous-réseau qu'elle (grâce au masque) et va pouvoir lui remettre directement (« * »).

Si la destination, par exemple un serveur de mails avec l'IP 93.56.134.23, n'est pas sur le même sous-réseau qu'elle, la machine va parcourir toute la table jusqu'à la fin. A la ligne *default* elle va trouver l'adresse de la passerelle qui sera censée transférer les paquets jusqu'à la destination.

3.6 Question 6

Le DNS (Domain Name System) permet de faire la correspondance entre une adresse IP et un nom d'hôte. Ainsi, une personne désirant accéder à un site internet saisira dans son navigateur `www.utt.fr` au lieu de 193.50.230.241. Ce qui est beaucoup plus facile à retenir.

Les correspondances insérées dans le fichier `/etc/hosts` sont consultées en premier par le système, ce qui peut donc poser des problèmes de sécurité si ces dernières redirigent de manière transparente l'utilisateur vers un site non désiré.

La ligne correspondant à la manière dont le système va résoudre les noms dans le fichier `nsswitch.conf` est :

```
⋮
hosts : files dns
⋮
```

files correspond au fichier `/etc/hosts` et *dns* désigne l'utilisation du resolveur. Ainsi, le système d'exploitation consultera dans un premier temps le fichier puis s'il n'a pas trouvé de correspondance, il utilisera le service DNS via son resolveur.

Mais le fichier `nsswitch.conf` comporte également les réglages pour d'autres bases de données telles que :

```
⋮
passwd : files ldap
shadow : files ldap
group : files ldap
⋮
```

passwd définit l'endroit où le système doit chercher les utilisateurs (ici en local et sur un annuaire), *shadow* est son équivalent mais pour les utilisateurs dont les mots de passe sont chiffrés. Enfin *group* désigne les emplacements où sont répertoriés les groupes auxquels les utilisateurs appartienent.

3.7 Outils de l'administrateur

Il existe de nombreux outils pour l'administrateur réseau, en voici quelques uns.

3.7.1 dig

Dig permet de tester la résolution DNS via un autre serveur que celui configuré sur l'hôte, mais également de consulter l'ensemble des enregistrements d'un nom de domaine. Ce qui est utile pour diagnostiquer un dysfonctionnement dans la résolution des noms.

Avec la commande ci-dessous, on effectue une recherche inversée permettant de retrouver la machine qui porte cette IP.

```
[root@server][~] dig -x 193.50.230.240 +short
pluton.utt.fr.
```

3.7.2 host

La commande host permet d'effectuer des requêtes DNS de manière simplifiée. On peut consulter les enregistrements pour un domaine précisé et ainsi vérifier le bon fonctionnement de son service DNS. Dans l'exemple suivant, la commande host nous indique l'ensemble des IPs enregistrées sous le nom de domaine *smtp-in.orange.fr*.

```
[root@server][~]  host  smtp-in.orange.fr
...
smtp-in.orange.fr has address 193.252.22.65
smtp-in.orange.fr has address 80.12.242.15
smtp-in.orange.fr has address 80.12.242.142
smtp-in.orange.fr has address 193.252.22.92
...
```

3.7.3 nmap

C'est un outil très complet, permettant de faire des analyses du réseaux pour retrouver les hôtes présents, les ports ouverts sur une machine, mais également de la détection de systèmes d'exploitations. Il permet donc de détecter l'ensemble des machines sur le réseaux ainsi que leurs failles potentielles.

Dans la capture suivante, on voit pour l'adresse IP scannée qu'il y a six ports ouverts, ainsi que leurs services associés.

```
[root@server][~] nmap 188.165.75.22
Host is up (0.063s latency).
Not shown: 992 filtered ports
PORT      STATE  SERVICE
22/tcp    open   ssh
25/tcp    open   smtp
80/tcp    open   http
110/tcp   open   pop3
143/tcp   open   imap
443/tcp   open   https
8080/tcp  closed http-proxy
10000/tcp closed snet-sensor-mgmt
```

3.7.4 traceroute

Traceroute permet de suivre le chemin qu'un paquet IP va prendre pour aller d'une machine A vers une machine B. Ce qui est utile pour vérifier que nos paquets prennent bien le chemin escompté.

Dans l'exemple suivant, on voit l'ensemble des machines par lesquelles un paquet IP est passé pour contacter google.fr.

```
[root@server][~] traceroute google.fr
traceroute to google.fr (209.85.229.104), 30 hops max, 40 byte packets
 1  ks305635.kimsufi.com (91.121.221.12)  0.000 ms  0.000 ms  0.000 ms
 2  rbx-63-m1.routers.ovh.net (91.121.221.253)  4.000 ms  0.000 ms  0.000 ms
 3  91.121.130.1 (91.121.130.1)  12.000 ms * *
 4  20g.ldn-1-6k.routers.chtix.eu (91.121.131.14)  4.000 ms * *
 5  195.66.224.125 (195.66.224.125)  4.000 ms  4.000 ms  4.000 ms
 6  64.233.175.27 (64.233.175.27)  8.000 ms  4.000 ms  8.000 ms
 7  72.14.232.134 (72.14.232.134)  8.000 ms  8.000 ms  8.000 ms
 8  ww-in-f104.1e100.net (209.85.229.104)  12.000 ms  12.000 ms  12.000 ms
```

3.7.5 netstat

La commande netstat permet d'afficher des statistiques sur les connexions réseaux, les ports en écoute et les sessions TCP établies.

Dans l'exemple ci-dessous, netstat permet de lister l'ensemble des ports en écoute sur le serveur, permettant ainsi de savoir quel service est en fonctionnement.

```
[root@server][~] netstat -atulpe
Connexions Internet actives (serveurs et établies)
Proto    Adresse locale          Adresse distante        Etat        User
tcp      *:ldap                  *:*                     LISTEN      root
tcp      *:sunrpc                *:*                     LISTEN      root
tcp      *:munin                 *:*                     LISTEN      root
tcp      *:ftp                   *:*                     LISTEN      proftpd
tcp      *:ssh                   *:*                     LISTEN      root
tcp      *:3128                  *:*                     LISTEN      root
```

3.7.6 ping

Ping est un outil simple mais très pratique. Il permet de vérifier si l'on peut contacter une machine, de connaître le temps moyen pour effectuer le parcours, mais aussi des statistiques telles que le nombres de paquets perdus, le nombre de routeurs traversés (via le ttl), ...

```
[root@server][~] ping www.yahoo.fr
PING rc.fy.b.yahoo.com (206.190.60.37) 56(84) bytes of data.
64 bytes from w2.rc.vip.re4.yahoo.com (206.190.60.37): icmp_seq=1 ttl=56 time=88.0 ms
64 bytes from w2.rc.vip.re4.yahoo.com (206.190.60.37): icmp_seq=2 ttl=56 time=88.0 ms
^C
--- rc.fy.b.yahoo.com ping statistics ---
2 packets transmitted, 2 received, 0\% packet loss, time 1004ms
rtt min/avg/max/mdev = 88.000/88.000/88.000/0.000 ms
```