TP1

Corrections

1 Interfaces réseau et Adresse IP

— Observez les interfaces `eth0` et `lo` de votre machine avec `/sbin/ifconfig`. Indiquez les adresses IPv4 et IPv6 correspondantes, la taille de la partie réseau, les tailles maximales de paquets (MTU, Maximum Transmission Unit), etc. Utilisez également la commande `ip addr ls` , c'est la même chose, mais en version plus moderne.

`ifconfig` retourne notamment :

```
eth0 Link encap:Ethernet HWaddr 00:13:72:4d:e3:c5
    inet adr:10.0.101.11 Bcast:10.0.101.255 Masque:255.255.255.0
    adr inet6: fe80::213:72ff:fe4d:e3c5/64 Scope:Lien
    adr inet6: 2001:660:6101:800:101::11/80 Scope:Global
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
    RX packets:20902080 errors:0 dropped:0 overruns:0 frame:0
    TX packets:39544505 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 lg file transmission:1000
    Octets reçus:3647472463 (3.6 GB) Octets transmis:2673409065 (2.6 GB)

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:1319655 errors:0 dropped:0 overruns:0 frame:0
    TX packets:1319655 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 lg file transmission:0
    Octets reçus:150697993 (150.6 MB) Octets transmis:150697993 (150.6 MB)
```

On repère pour l'interface `eth0` *une adresse IPv4* `10.0.101.11`*, pour laquelle le masque est* `255.255.255.0`*, donc un /24, et deux adresses IPv6* `2001:660:6101:800:101::11/80` *et* `fe80::213:72ff:fe4d:e3c5/64` *(ip link pour réseau local sans routage, ip global pour adresse publique). La taille maximale des paquets est 1500 octets.*
Pour l'interface `lo` *l'adresse IPv4 est* `127.0.0.1`*, le masque* `255.0.0.0`*, l'adresse IPv6 est* `::1/128`*, la taille maximale est 16436 octets.*
En IPv6, l'écriture est en hexadécimale sur 16 octets, où les 8 groupes de 2 octets (soit 16 bits par groupe) sont séparés par un signe deux-points :
`2001:0db8:0000:85a3:0000:0000:ac1f:8001`
Il est permis d'omettre de 1 à 3 chiffres zéros non significatifs dans chaque groupe de 4 chiffres hexadécimaux. Ainsi, l'adresse IPv6 ci-dessus est équivalente à :
`2001:db8:0:85a3:0:0:ac1f:8001`
De plus, une unique suite de un ou plusieurs groupes consécutifs de 16 bits tous nuls peut être omise, en conservant toutefois les signes deux-points de chaque côté de la suite de chiffres omise, c'est-à-dire une paire de deux-points (: :)1. Ainsi, l'adresse IPv6 ci-dessus peut être abrégée en :
`2001:db8:0:85a3::ac1f:8001`
En revanche l'écriture suivante n'est pas valide
`2001:db8::85a3::ac1f:8001`
car elle contient plusieurs substitutions (dont les longueurs binaires respectives sont ici ambiguës) : il ne peut exister qu'une seule occurrence de la séquence : : dans la notation d'une adresse IPv6.
`ip addr ls` *retourne notamment :*

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host
valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
link/ether 80:c1:6e:f2:e7:84 brd ff:ff:ff:ff:ff:ff
inet 10.0.101.11/24 brd 10.0.101.255 scope global eth0
valid_lft forever preferred_lft forever
inet6 2001:660:6101:800:101::11/80 scope global
valid_lft forever preferred_lft forever
inet6 fe80::82c1:6eff:fef2:e784/64 scope link
valid_lft forever preferred_lft forever
```

.

— D'après ses adresses IPv4 et IPv6, à quoi correspond l'interface `lo` ?

On a vu en cours que ses adressees IP, `127.0.0.1` et `::1/128`, sont spéciales : elle désigne "moi-même". C'est donc une interface réseau qui permet aux programmes tournant sur une même machine de se connecter les uns aux autres aussi, sans passer par le réseau physique externe.

.---

— Observez que la MTU n'est pas la même, pourquoi ?

Une carte réseau a des tampons de taille limitée (qui est la raison pour laquelle TCP découpe de toutes façons nos données en morceaux avant de les envoyer), ici 1500. L'interface `lo`, puisqu'elle n'est pas une vraie interface physique, n'a pas de telle limitation.

.---

— La commande `ping -4 uneIP` permet de tester la connectivité IPv4 par l'émission d'une requête echo ICMP. Vérifiez que vous êtes bien relié à la machine de votre voisin. De même, avec `ping6 uneIPv6` pour les adresses IPv6.

La sortie de la commande `ifconfig` du voisin indique que son IP est `10.0.101.9`, en utilisant `ping 10.0.101.9`, on obtient :

```
PING 10.0.101.9 (10.0.101.9) 56(84) bytes of data.
64 bytes from 10.0.101.9: icmp_seq=1 ttl=64 time=2.41 ms
64 bytes from 10.0.101.9: icmp_seq=2 ttl=64 time=0.138 ms
```

L'autre machine répond effectivement bien. De même en IPv6.

```
PING 2001:660:6101:800:101::10(2001:660:6101:800:101::10) 56 data bytes
64 bytes from 2001:660:6101:800:101::10: icmp_seq=1 ttl=64 time=0.626 ms
64 bytes from 2001:660:6101:800:101::10: icmp_seq=2 ttl=64 time=0.287 ms
```

.---

2 Protocole ARP

— Identifiez les adresses des machines voisines avec lesquelles des échanges récents ont eu lieu (table ARP, disponible par la commande `/usr/sbin/arp` , on peut utiliser l'option `-n` pour avoir les adresses IP plutôt que les noms de machines). Il doit y avoir au moins l'adresse du routeur (en `.254`, on verra dans la section routage ci-dessous).

arp donne effectivement

```
Address                  HWtype  HWaddress           Flags Mask  Iface
glenfiddich.emi.u-borde  ether   00:13:72:54:b2:7f   C           eth0
10.0.101.254             ether   00:18:18:e8:0f:e0   C           eth0
```

glenfiddich est effectivement la machine de mon voisin, et 10.0.101.254 a l'air d'être l'adresse IP du routeur pour notre salle (ici la 101). On a sur les mêmes lignes leurs adresses MAC.

.

— Vérifiez que lorsque vous émettez avec ping une requête ICMP echo vers une machine qui ne figure pas encore dans votre table ARP, cette machine apparaît dans la table ARP de votre machine et la vôtre, dans la table ARP de l'autre.

La machine de mon autre voisin, 10.0.101.7 (caolila), n'était pas dans la table ARP. Après avoir tapé ping 10.0.101.7, relancer la commande arp montre effectivement l'entrée dans la table avec son adresse MAC :

```
Address                  HWtype  HWaddress           Flags Mask  Iface
glenfiddich.emi.u-borde  ether   00:13:72:54:b2:7f   C           eth0
caolila.emi.u-bordeaux1  ether   00:26:b9:75:bb:3e   C           eth0
10.0.101.254             ether   00:18:18:e8:0f:e0   C           eth0
```

Si je demande à mon voisin de regarder dans sa table ARP, ma machine y est apparue (même si mon voisin n'a pas lui-même lancé de ping vers ma machine, il a bien fallu que sa machine envoie sa réponse à la mienne).

.

— Essayez la commande `ip neigh ls` ; c'est la même chose en version plus moderne, et contient notamment aussi les voisins en IPv6.

```
ip neigh ls
fe80::82c1:6eff:fef2:e80d dev eth0 lladdr 80:c1:6e:f2:e8:0d STALE
fe80::ae16:2dff:fe00:4930 dev eth0 lladdr ac:16:2d:00:49:30 STALE
2001:660:6101:800:101::12 dev eth0 lladdr ac:16:2d:00:49:30 STALE
2001:660:6101:800:101::10 dev eth0 lladdr 80:c1:6e:f2:e8:0d STALE
fe80::218:18ff:fee8:fcb dev eth0 lladdr 00:18:18:e8:0f:cb router STALE
10.0.101.9 dev eth0 FAILED
10.0.101.254 dev eth0 lladdr 00:18:18:e8:0f:cb REACHABLE
10.0.101.10 dev eth0 lladdr 80:c1:6e:f2:e8:0d STALE
10.0.101.12 dev eth0 lladdr ac:16:2d:00:49:30 STALE
```

.

3 Résolution de noms (DNS)

Mais au fait, comment la traduction `www.yahoo.com` → 98.139.180.149 fonctionne ? À l'aide d'un serveur DNS.

— Lisez le fichier `/etc/resolv.conf` , pourquoi y a-t-il plusieurs adresses IP ?

```
 cat /etc/resolv.conf
nameserver 10.0.220.13
nameserver 10.0.220.14
search emi.u-bordeaux.fr
domain emi.u-bordeaux.fr
```

Pour la redondance : si l'un des deux serveurs tombe en panne, l'autre peut répondre pendant que l'on répare le premier.

.

La ligne `search` permet d'éviter d'avoir à taper le nom de machine en entier. Essayez

de taper `https://services/` tout court dans un navigateur web et observez comment cela est complété pour confirmer.

Cela sert de raccourci : on peut par exemple donner à `ping` le nom de la machine du voisin, ici `emi.u-bordeaux1.fr` sera automatiquement ajouté.

```
 ping services
PING dumbledore.emi.u-bordeaux1.fr (10.0.252.5) 56(84) bytes of data.
64 bytes from dumbledore.emi.u-bordeaux1.fr (10.0.252.5): icmp_seq=1 ttl=63
time=0.777 ms
```

.

— Pour effectuer explicitement une résolution de nom, utilisez la commande `host` (ou éventuellement `nslookup`).

— Essayez de résoudre `www.yahoo.com` . Pourquoi y a-t-il plusieurs adresses IP ?

```
host www.yahoo.com
www.yahoo.com is an alias for fd-fp3.wg1.b.yahoo.com.
fd-fp3.wg1.b.yahoo.com has address 98.139.180.149
fd-fp3.wg1.b.yahoo.com has address 98.139.183.24
fd-fp3.wg1.b.yahoo.com has IPv6 address 2001:4998:58:c02::a9
```

```
nslookup www.yahoo.com
Server: 127.0.1.1
Address: 127.0.1.1#53

Non-authoritative answer:
www.yahoo.com canonical name = fd-fp3.wg1.b.yahoo.com.
Name: fd-fp3.wg1.b.yahoo.com
Address: 98.139.180.149
Name: fd-fp3.wg1.b.yahoo.com
Address: 98.139.183.24
```
De nouveau, pour la redondance, il y a plusieurs serveurs web qui peuvent répondre, si le premier tombe les autres seront essayés.

.---

Observez également que cela retourne à la fois des adresses IPv4 et une adresse IPv6. Réessayez plusieurs fois. Il peut arriver que le résultat soit différent, pourquoi ?

Pour répartir la charge.

.---

— Parfois il est aussi utile d'ajouter des noms de machine à la main, lisez le fichier `/etc/hosts` (et le manuel).

On a effectivement quelques entrées dans /etc/hosts :
`127.0.0.1 localhost virtual-fai`
`10.0.230.12 firenze.emi.u-bordeaux1.fr firenze`
`10.0.220.5 eta.emi.u-bordeaux1.fr eta`
127.0.0.1 c'est la machine elle-même, en local. firenze et eta sont a priori des serveurs.
`man hosts`
""
Sur les systèmes modernes, même si la table des hôtes a été remplacée par DNS, ce mécanisme est encore largement employé pour initialiser une machine. La plupart de systèmes ont un petit fichier contenant le nom et l'adresse des hôtes importants sur le réseau local. C'?est utile lorsque le DNS n'est pas actif, notamment lors de la mise en route du système.
""
`/etc/nsswitch.conf` indique notamment `hosts: files dns`, donc c'est le fichier `/etc/hosts` qui prend le pas sur la résolution DNS.

Les quelques entrées mises à la main dans `/etc/hosts` *peuvent être utiles pour que les machines continuent à fonctionner (NFS notamment) même si le serveur DNS tombe en panne.*

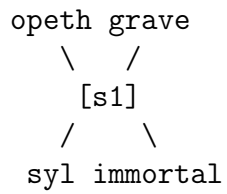
.

---

4 Configuration d'un réseau local

Vous allez utiliser un environnement permettent d'émuler un réseau de machines virtuelles (ou VMs). L'environnement que nous allons utiliser est *QemuNet*. Cet environnement s'appuie entièrement sur le projet *Qemu* pour émuler des machines virtuelles et sur *VDE* pour émuler des switchs Ethernet. Les VMs que nous utilisons sont des architectures x86_64 sous Linux Debian 8. L'intérêt principal de cette approche tient au fait que vous pouvez tester facilement des topologies réseaux diverses. Par ailleurs, il vous est possible de disposer d'un compte administrateur (*root* sous Linux) sur les VMs, sans risquer de ne rien casser !

Voici la configuration réseau utilisée pour le TP1, un réseau local (LAN) de 4 machines interconnectés via le switch Ethernet *s1*.



Afin de démarrer ce réseau virtuel au CREMI, il convient de taper la commande suivante :
`/net/ens/qemunet/qemunet.sh -x -s /net/ens/qemunet/demo/lan0.tgz`
Une fois le script lancé, vous disposez pour chaque VM d'un simple terminal en mode texte : c'est bien suffisant pour faire tout ce que nous voulons... Il n'est donc pas question de lancer un programme graphique ! Comme éditeur de texte, vous disposez au besoin de *nano* ou *jed*[1].

— Une fois que les 4 terminaux des VM sont apparus (Attention, les 4 fenêtres peuvent être empilées) : connectez-vous en tant que root sur immortal (sans password).
— A l'aide de la commande ifconfig (man ifconfig), donnez la liste des interfaces réseaux.

Faire `ifconfig -a` *pour voir toutes les interfaces...*

1. Un clône léger de Emacs qui dispose de raccourcis clavier similaires : c-x c-f (open file), c-x c-s (save file), c-x c-c (exit), ...

.

— On décide de configurer l'interface eth0 de telle sorte que la machine *immortal* possède l'adresse `192.168.0.1/24`. Quelle est l'adresse du réseau ? Quel est le masque du réseau ? Quelle est la plage d'adresse IP de ce réseau.

L'adresse du réseau 192.168.0.0 et l'adresse de diffusion est 192.168.0.255. Le masque /24 s'écrit également 255.255.255.0. La plage d' IP du réseau est 192.168.0.0-255.

.

— Configurez immortal à l'aide de la commande ifconfig. Configurez de manière analogue les 3 autres machines.

immortal# ifconfig eth0 192.168.0.1/24 up
On donne aux 3 autres machines une adresse IP dans le même réseau, par exemple 192.168.0.2/24, .3/24 et .4/24. Et c'est tout !

.

— Vérifiez vos configurations à l'aide de la commande ping. Quel est le protocole utilisé par le programme ping ?

Il faut utiliser les IPs explicitement, car nous n'avons pas de serveur DNS pour résoudre les noms immortal, opeth, ...
opeth# ping 192.168.0.1

.

— Mettez en évidence que le ping fonctionne à l'aide de la commande `tcpdump -i eth0` qui permet d'afficher tout le trafic réseau entrant et sortant d'une certaine machine (sur l'interface eth0).

opeth# ping 192.168.0.1
immortal#tcpdump -i eth0
20:55:37.145246 IP 192.168.0.4 > 192.168.0.1: ICMP echo request, id 565, seq 17, length 64
20:55:37.145281 IP 192.168.0.1 > 192.168.0.4: ICMP echo reply, id 565, seq 17, length 64

.

— Essayez un ping avec l'adresse de broadcast du réseau. Que se passe-t-il ? Les requettes sont elles reçues par toutes les machines ? Est-ce qu'elles y répondent ? Corrigez le problème en tapant cette commande sur tous les postes :
`echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts`
Re-essayer un ping avec l'adresse de broadcast du réseau. Que constatez-vous ?
— Avec la commande *reboot*, redémarrez la machine *immortal*. Vous notez que l'interface réseau a perdu sa configuration ! Pour remédier à ce problème, il faut editer le fichier `/etc/network/interfaces` et y donner la configuration de l'interface eth0 : `iface eth0 inet static`. Cherchez sur Internet et dans le man (`man interfaces`) comment configurer ce fichier. N'oubliez pas de mettre une ligne `auto eth0` Notez que ce fichier est interprété seulement au démarrage de la machine, ou lorsque vous appelez explictement le script `/etc/init.d/networking restart`. Testez votre configuration.

Fichier de configuration /etc/network/interfaces :

```
# loopback
auto lo
iface lo inet loopback

# interface eth0
auto eth0
iface eth0 inet static
address 192.168.0.1
netmask 255.255.255
```

Ne pas oublier auto eth0*, sinon l'interface ne sera pas chargé automatiquement ! Puis lancez :* `/etc/init.d/networking restart` *avant de faire un* `reboot`*. Les erreurs sont consultables à la fin du fichier /var/log/syslog.*

.

— Clôturez votre session *QemuNet*. Pour ce faire, vous pouvez fermer toutes les VMs proprement avec la commande *poweroff* ou plus simplement faire un `Ctrl-C` dans le terminal où vous avez démarré votre session.
— Bonus : Configurez le réseau en IPv6 en découpant le réseau `2001:db8::/48` en sous-réseaux.

C'est tout pareil, juste avec des ipv6 :)
Juste un détail pour IPv6 : comme il y a déjà une adresse en fe80 : :, il faut demander un ajout :
`ifconfig eth0 inet6 add 2001:db8:1::1/64`
L'adresse de broadcast est par contre indépendante du réseau, c'est toujours `ff02::1`*.*

```
# interface eth0
auto eth0
iface eth0 inet6 static
address 2001:db8:1::1
netmask 64
```

.
