

# TP1

## 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.
- D'après ses adresses IPv4 et IPv6, à quoi correspond l'interface `lo` ?
- Observez que la MTU n'est pas la même, pourquoi ?
- 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.

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

## 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 ?  
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.

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

## 4 Configuration d'un réseau local

Vous allez utiliser un environnement permettant 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*.

```

opeth grave
  \   /
   [s1]
  /   \
syl immortal

```

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*<sup>1</sup>.

- 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.
- 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.

---

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), ...

- Configurez *immortal* à l'aide de la commande `ifconfig`. Configurez de manière analogue les 3 autres machines.
- Vérifiez vos configurations à l'aide de la commande `ping`. Quel est le protocole utilisé par le programme `ping` ?
- 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`).
- Essayez un `ping` avec l'adresse de broadcast du réseau. Que se passe-t-il ? Les requêtes 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 éditer 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 explicitement le script `/etc/init.d/networking restart`. Testez votre configuration.
- 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.