TP2

Corrections

Tous les protocoles au dessus d'IP sont numérotés, comme TCP ou UDP. Une table est lisible dans `/etc/protocols`, trouvez-y les numéros de TCP et UDP. Les services TCP (http, ftp, ssh, ...) sont également numérotés (et sont appelés **numéros de port**), la table est dans `/etc/services`, trouvez-y http, ftp, ssh, ... Les numéros entre 0 et 1023 ne sont utilisables que par l'utilisateur `root`.

On voit effectivement dans `/etc/protocols` que le numéro de protocole d'UDP est 17, et celui de TCP est 6. Le numéro de port de www est effectivement 80.

1 Netcat

— Lancez `nc -l -p 12345` et laissez-le tourner . Observez (en passant par un autre terminal) dans `netstat -Ainet -Ainet6 -a` l'apparition du service parmi les autres.

On voit effectivement apparaître

```
Proto  Recv-Q  Send-Q  Adresse locale  Adresse distante  Etat
tcp         0       0  *:12345         *:*               LISTEN
```

— Ajoutez à netstat l'option `-p` pour constater que c'est bien le programme `nc` qui est à l'écoute.

En effet, on obtient une colone de plus :

```
Proto  Recv-Q  Send-Q  Adresse locale  Adresse distante  Etat    PID/Program name
tcp         0       0  *:12345         *:*              LISTEN  5963/nc
```

— Dites à votre voisin de lancer `nc votremachine 12345` . Observez dans `netstat -Ainet -Ainet6` la connexion établie.

On voit effectivement apparaître

```
Proto  Recv-Q  Send-Q  Adresse locale      Adresse distante   Etat
tcp         0       0  glenmorangie:12345  glenfiddich:47334  ESTABLISHED
```

D'un côté, et

```
Proto  Recv-Q  Send-Q  Adresse locale      Adresse distante   Etat
tcp         0       0  glenfiddich:473345  glenmorangie:1234  ESTABLISHED
```

de l'autre.

— Tapez des lignes d'un côté ou de l'autre, observez que c'est effectivement transmis de l'autre côté.
— Comment transmettre un fichier d'une machine à l'autre à l'aide de `nc` et de redirection bash ? (utilisez l'option `--send-only` côté émetteur)

Lancer par exemple d'abord le côté réception :

```
nc -l -p 12345 > lefichier
```

puis le côté émission :

```
nc -send-only votremachine 12345 < lefichier
```

---

2 Services au CREMI : LDAP & NFS

Quelle que soit la machine que vous utilisez au CREMI, elle vous connaît et retrouve vos fichiers. Comment cela se passe-t-il ?

— Pour l'identification et l'authentification, c'est le protocole LDAP qui est utilisé, la configuration est lisible dans le fichier `/etc/ldap.conf` (pas la peine de tout lire, ce qui nous intéresse est au tout début), pourquoi y a-t-il plusieurs serveurs (`host`) ?

```
cat /etc/ldap.conf | head
#host cresus dionysos bromios
host cresus dionysos bromios

base DC=cremi,DC=emi,DC=u-bordeaux1,DC=fr
ldap_version 3

# Optional: default is 389.
#port 389
```

Encore une fois, pour la redondance. Si le serveur LDAP tombait en panne, plus aucune machine au CREMI n'accepterait d'utilisateur !

.---

Trouvez le numéro de port de ce service, vérifiez dans `/etc/services`.

```
$ grep ldap /etc/services
ldap 389/tcp # Lightweight Directory Access Protocol
ldap 389/udp
ldaps 636/tcp # LDAP over SSL
ldaps 636/udp
```

Il y a apparemment une version normale sur le port 389, et une version "sécurisée" par SSL sur le port 636.

.---

— Vos fichiers sont stockés sur un serveur NFS, utilisez la commande `df ~` pour repérer le nom du serveur, et chemin sur le serveur.

On obtient

```
Sys. de fichiers blocks de 1K Utilisé Disponible Uti% Monté sur
netapp:/vol/account_cremi 2147483648 1065841792 1081641856 50% /autofs/netapp/
account/cremi
```

Le serveur s'appelle netapp, et le chemin est `/vol/account_cremi`

.

3 Connexion à une machine distante avec SSH

Utilisez la commande `ssh -X` pour vous connecter sur la machine du voisin. Lancez `xeyes`, sur quelle machine s'exécute-t-il ?

En utilisant `ps`, on se rend compte que `xeyes` s'exécute sur la machine du voisin, même si l'affichage se retrouve sur la machine locale.

.

Comment l'affichage se retrouve-t-il sur votre machine ?

Via la connexion ssh.

.

4 Routage

4.1 Préliminaires

Pour dialoguer avec la Terre entière, il est nécessaire de passer par des passerelles. Lancez la commande `/sbin/route -n` . Il y a deux routes importantes.

- La route des machines de la salle (en `10.0.x.y`), qui sont accessibles directement en Ethernet sans passer par un routeur, remarquez le `Genmask`, il indique la séparation entre la partie réseau (bits à 1) et la partie machine des adresses IP (bits à 0). Comparez avec les adresses de vos voisins : êtes-vous bien dans le même réseau ?
- La route par défaut (`0.0.0.0`), qui utilise une passerelle (quelle est son adresse IP ?)

La commande `/sbin/route` retourne

Destination	Passerelle	Genmask	Indic	Metric	Ref	Use	Iface
10.0.101.0	*	255.255.255.0	U	0	0	0	eth0
default	10.0.101.254	0.0.0.0	UG	100	0	0	eth0

10.0.101.254 est donc bien l'adresse de la passerelle : pour toute les destinations en 10.0.101.x, on n'a pas besoin de passerelle, elles sont directement accessible par eth0. Pour le reste, on doit passer par la passerelle.

.---

On peut aussi utiliser la version plus moderne `ip route ls` . À quoi correspond le suffixe `/24` ?

```
   ip route ls
default via 10.0.101.254 dev eth0
10.0.101.0/24 dev eth0 proto kernel scope link src 10.0.101.11
172.16.0.0/24 dev tap0 proto kernel scope link src 172.16.0.2
172.16.1.0/24 dev tap1 proto kernel scope link src 172.16.1.2
```

/24 correspond aux 24 premiers bits à 1 de 255.255.255.0, le masque séparant la partie réseau de la partie machine.

.---

Pour observer en IPv6, on utilise `ip -6 route ls` . On a les mêmes deux routes importantes :

- La route des machines de la salle (en `2001:660:6101:800:x::y`), qui sont accessibles directement en Ethernet. Observez le suffixe `/80`, et comptez le nombre de chiffres hexadécimaux auxquels cela correspond dans l'adresse IP pour déterminer la partie réseau et la partie machine. (attention, les 0 non significatifs ne sont pas écrits en IPv6, chaque paquet séparé par `:` compte pour 16 bits).
- La route des adresses locales `fe80::/64` : on a sur `eth0` non seulement une adresse publique (`2001:...`), mais aussi une adresse locale en `fe80::/64`. L'une ou l'autre sera utilisée selon que l'on veut émettre en local seulement, ou bien sur le reste d'Internet.
- La route par défaut (`default`) utilise une passerelle de type `fe80::...` : c'est l'adresse IPv6 locale du routeur pour cette salle-ci.

```
ip -6 route ls
2001:660:6101:800:101::/80 dev eth0 proto kernel metric 256 expires 2591903sec
fe80::/64 dev eth0 proto kernel metric 256
fe80::/64 dev br0 proto kernel metric 256
fe80::/64 dev tap0 proto kernel metric 256
default via fe80::218:18ff:fee8:fcb dev eth0 proto ra metric 1024 expires 8903sec
```

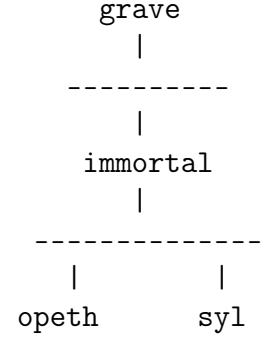
Memento Routage

En guise de documentation rapide, voici un résumé des commandes que l'on va utiliser dans les sections suivantes. Il n'y a donc pas d'action à faire dans cette section, c'est juste de la documentation. Il faut bien sûr remplacer `<@gateway>` par une adresse IP, etc.

- Activer le relai des paquets sur une machine (ip forward) :
  `echo 1 > /proc/sys/net/ipv4/ip_forward`
- Configurer de manière permanente le relai des paquets : voir le fichier `/etc/sysctl.conf`
- Afficher la table de routage : `route -n`
- Définir une route par défaut : `route add default gw <@gateway>`
- Ajouter une route vers un réseau spécifique :
  `route add -net <@network/bits> gw <@gateway>`
- Pour supprimer une règle, il faut taper la commande `route del <...>` avec exactement les mêmes arguments que pour la commande `add`.
- Vérifier la route envoyer un paquet à une adresse spécifique : `traceroute <@host>`
- On peut aussi utiliser la commande `ip` pour ajouter des route :
  `ip route add <@network/bits> via <@gateway>` (de même pour une seule adresse).

4.2 Routage Basique

Dans un LAN, toutes les machines peuvent communiquer directement, car elles sont physiquement connectées par leurs interfaces réseaux. Dans un réseau plus complexe, comme celui que nous allons étudier maintenant, il est nécessaire de configurer les tables de routage des machines, pour qu'elles collaborent à l'acheminement des messages d'un bout à l'autre du réseau.



Pour lancer la topologie ci-dessus, veuillez taper la commande suivante sur votre propre machine :

`/net/ens/qemunet/qemunet.sh -x -s /net/ens/qemunet/demo/gw1.tgz`

S'assurer que c'est bien lancé directement sur la machine, et non sur une autre machine via ssh (auquel cas l'accélération matérielle ne fonctionne pas).

.

- Les adresses IPs sont déjà configurées. Veuillez reporter chaque adresse IP sur le schéma ci-dessus en précisant à chaque fois le nom de l'interface ethX. Pour vous aider, vous pouvez consulter, sur la machine hôte, le fichier `demo/gw.topo` ce fichier détaille la configuration du réseau virtuel dans QemuNet. Puisque certaines machines ont plusieurs adresses IP, il vaut mieux systématiquement donner aux commandes des adresses IP, et non des noms de machines.
- Vérifiez avec 'ping' que les machines peuvent communiquer dans leurs réseaux locaux respectifs.
- Afficher les tables de routage avec la commande 'route -n'.
- Configurez les tables de routage des différentes machines à l'aide de la commande 'route'. Il faut également activer le relai des paquets sur `immortal` pour qu'il agisse comme un routeur. On rappelle quelques éléments de syntaxe dans le memento ci-dessous. Vérifiez avec ping que toutes les machines sont capables de communiquer ensemble. Si ça ne passe pas, utilisez `tcpdump -n -i any` sur les différentes machines pour voir où ça coince.
- Faites un ping entre opeth et grave. Lancez `tcpdump -n -i any` sur immortal afin d'afficher le trafic qui circule...
- Si vous ne l'avez pas fait, simplifiez le routage en utilisant `default` plutôt que des routes explicites.

---

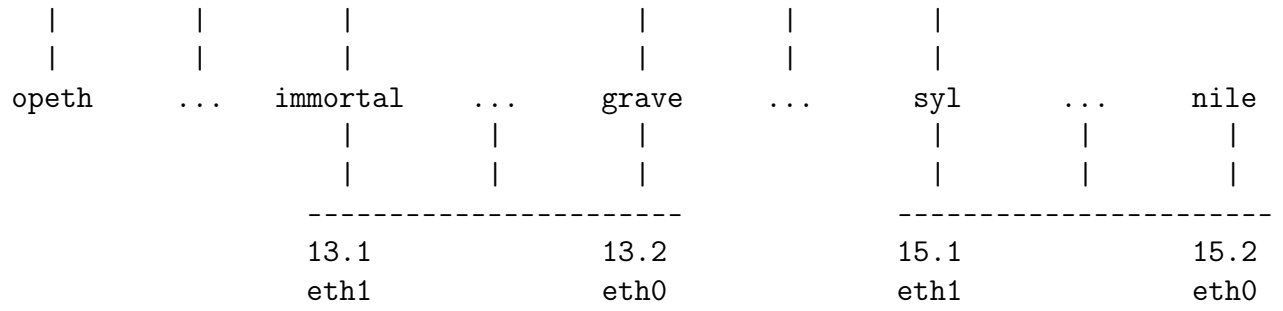
grave$ route add default gw 147.210.0.1
immortal$ echo 1 > /proc/sys/net/ipv4/ip_forward
opeth$ route add default gw 192.168.0.1
syl$ route add default gw 192.168.0.1

.

4.3 Routage Avancé

Voici une nouvelle configuration, composée de 4 sous-réseaux /24 dans le réseau 147.210.0.0/16 :

```
 eth0            eth0            eth1            eth0
 12.1            12.2            14.1            14.2
----------------------          ----------------------
```



Démarrez ce réseau virtuel :

```
/net/ens/qemunet/qemunet.sh -x -s /net/ens/qemunet/demo/chain0.tgz
```

Les adresses IP sont déjà configurées. Il faut donc configurer les tables de routage afin que tout le monde puissent communiquer avec tout le monde. La machine grave nécessite l'utilisation de routes spécifiques, les autres peuvent utiliser des routes `default`

---

grave$ echo 1 > /proc/sys/net/ipv4/ip_forward
grave$ route add -net 147.210.12.0/24 gw 147.210.13.1
grave$ route add -net 147.210.15.0/24 gw 147.210.14.2
immortal$ echo 1 > /proc/sys/net/ipv4/ip_forward
immortal$ route add default gw 147.210.13.2
nile$ route add default gw 147.210.15.1
opeth$route add default gw 147.210.12.2
syl$ echo 1 > /proc/sys/net/ipv4/ip_forward
syl$ route add default gw 147.210.14.1

.---

5 Bonus : Topologie réseau

Connectez-vous à la machine de votre voisin avec `ssh leNomDeLaMachine` . Lancez `ip neigh ls` pour afficher le cache ARP/NDP sur chacune des deux machines, constatez que chacune a une entrée pour le voisin.

Connectez-vous maintenant à `jaguar`, lancez `ip neigh ls` , pourquoi l'adresse MAC de votre machine n'y apparait pas ?

Parce que jaguar est sur un autre réseau et doit passer par la passerelle de son réseau à lui (10.0.252.254 et fe80 : :218 :19ff :fe00 :3e8a, visibles dans `ip neigh ls`*, elles) pour envoyer des paquets à ma machine.*

.

Essayez de déterminer la topologie réseau. On a vu en cours que les machines de salle de TP avaient des adresses similaires. Déterminez comment sont regroupées les IPs serveurs (`bacchus`, `dns0`, `dns1`, `cocatrix`, `infini1`, `jaguar`, `mcgonagall`, `netapp`), pour faire un petit schéma du réseau

---

On observe trois groupes :

- *`bacchus`, `dns0`, `dns1`, `netapp` sont en 10.0.220.x. `bacchus` était le serveur windows, `netapp` était le serveur NFS, `dns0` et `dns1` sont les serveurs DNS, ce sont donc a priori les serveurs.*
- *`cocatrix`, `infini1`, `mcgonagall` sont en 10.0.230.x. `mcgonagall` est un serveur de travail, de même que `hagrid` et `infini1`.*
- *`jaguar` est à part en 10.0.252.x. C'est en effet la seule machine visible depuis l'extérieur.*

.

Observez les routes configurées sur les serveurs auxquels vous avez accès `ssh`, contentez-vous d'observer l'adresse IP pour les autres. Qu'en déduire sur la topologie réseau ? Pourquoi est-il organisé ainsi ?

Les serveurs en 10.0.230.x utilisent la passerelle 10.0.230.254. Jaguar est seule et utilise la passerelle 10.0.252.254. Apparemment ces machines sont donc regroupées dans des réseaux distincts, selon l'utilisation qui en est faite, sans doute pour simplifier la mise en place de règles de firewall.

.
