Enoncé du TP 6 Réseaux

Commandes réseaux

C. Pain-Barre

INFO - IUT Aix-en-Provence

version du 16/4/2012

---

1 Configuration d'un hôte sous Unix

1.1 Les interfaces réseau

Une interface réseau identifie un périphérique permettant de se connecter à un réseau ainsi que les méthodes d'accès à ce réseau. Ce peut être un modem, une carte réseau, un port série, un port USB, ou autre. Une station ne possède et n'utilise généralement qu'une seule carte réseau. Les routeurs possèdent en revanche une interface par réseau auquel ils sont connectés.

Sur Unix (et Linux), une interface correspond à un point d'entrée dans le noyau (cœur du système). Envoyer des messages via les interfaces réseaux revient à passer des données à des procédures spéciales du noyau chargées d'effectuer les opérations d'entrées-sorties physiques. Une interface est généralement identifiée par un nom logique indiquant le type d'interface et le numéro d'ordre de la carte. Par exemple, sous Linux, une carte Ethernet classique sera identifiée par :

- **`eth0`** pour la première carte ;
- **`eth1`** pour la seconde ;
- etc.

et les cartes Wifi, plutôt par **`wlan0`**, **`wlan1`**, ...

Sur SunOS 7.0 (Unix de Sun MicroSystems), une carte Ethernet sera identifiée par :

- **`le0`** pour la première ;
- **`le1`** pour la seconde ;
- etc.

> ✍ Une exception concerne l'interface ***loopback*** identifiée par **`lo`** suivi éventuellement d'un numéro. Cette interface correspond aux adresses IP commençant par **`127`**. La plus communément utilisée étant l'adresse **`127.0.0.1`**. Généralement, les stations sont configurées pour que cette interface puisse être désignée par le nom **`localhost`** (cas des stations Unix et Windows). L'interface ***loopback*** n'est pas rattachée à une carte réseau. C'est en fait une adresse permettant de tester en local uniquement des programmes utilisant TCP/IP, sans même disposer d'une liaison réseau. Cela permet donc de réaliser des tests sans pour autant provoquer de transmission sur le réseau, ou d'utiliser localement des services réseaux.

Une interface possède (généralement) une adresse physique, communément appelée adresse MAC. C'est cette adresse qui est utilisée pour les communications au niveau trame (couche liaison OSI), dans le réseau physique (local). Pour que cette interface puisse être une destination dans l'Internet, il faut lui associer une adresse IP (une seule suffit). Ainsi, une station va posséder une seule adresse IP alors qu'un routeur va posséder une adresse IP par réseau auquel il est connecté, via une interface. Pour réaliser cette association, il faut configurer l'interface. La configuration d'une interface comprend :

- l'adresse IP qui lui sera associée ;
- le masque de sous-réseau ;

> ✍ l'adresse du réseau de l'hôte est déduite en appliquant le masque à l'adresse IP.

- selon le système et le contexte, l'adresse d'un routeur (*Gateway*) pour créer la route par défaut ;
- l'adresse IP de diffusion dirigée (si possible) dans le réseau de l'hôte. Un datagramme envoyé vers cette adresse est aussi destiné à cette interface ainsi qu'à toutes celles raccordées à ce réseau ;

> ✍ l'adresse de diffusion dirigée est aussi déduite de l'adresse IP et du masque, en mettant à 1 dans l'adresse IP, les bits qui sont à zéro dans le masque.

- un état actif (up) ou inactif (down) ;
- un certain nombre d'options :
  - ◇ le MTU (Maximum Transmission Unit) : charge utile maximale d'une trame émise via cette interface, dépendant du réseau. Vaut 1500 pour Ethernet ;
  - ◇ l'activation ou non du protocole ARP sur cette interface ;
  - ◇ la possibilité de diffuser ou non via l'interface (BROADCAST) ;
  - ◇ la possibilité de recevoir des messages émis en multi-diffusion (MULTICAST) ;
  - ◇ l'activation du mode ***promiscuous***, donnant la possibilité de recevoir toutes les trames émises sur le réseau, même celles n'étant pas destinées à l'adresse physique de cette interface. Ce mode est utilisé pour réaliser des captures de trafic réseau ;

1.2 ifconfig : configuration d'une interface sous Unix

La commande permettant de configurer une interface sous Unix est **ifconfig** (**i**nter**f**ace **config**uration), qui se trouve dans le répertoire `/sbin`. Elle admet d'assez nombreuses options car elle permet de configurer tout type de carte, pour différents besoins et protocoles réseau (pas seulement IPv4). Elle sert aussi à la consultation de la configuration courante des interfaces.

> ✍ Le répertoire `/sbin` tout comme `/usr/sbin` contiennent des commandes destinées normalement aux administrateurs. Ces répertoires ne sont donc pas contenus par défaut dans le **`PATH`** des utilisateurs. Un utilisateur normal voulant les utiliser doit soit saisir leur référence absolue soit modifier son **`PATH`** en tapant :
>
> ```
> $ PATH="$PATH:/sbin:/usr/sbin"
> ```

Synopsis

`ifconfig` [**`-a`** | *interface*]
`ifconfig` *interface* [*adresse-ip*] [**`netmask`** *masque*] [**`mtu`** *mtu*] [**`hw ether`** *adresse-mac*]

La première forme sert à la consultation de la configuration courante. Si *interface* n'est pas indiquée, seule la configuration des interfaces actives (UP) est affichée. Les interfaces inactives (DOWN) sont aussi affichées si on utilise **`-a`**. Sinon, seule la configuration de *interface* est affichée.

La deuxième forme permet une configuration élémentaire de l'*interface* indiquée, où :

- *adresse-ip* est l'adresse IPv4 à lui attribuer ;
- **netmask** *masque* précise le masque de sous-réseau à utiliser ;
- **mtu** *mtu* précise le MTU (en octets). Il n'est généralement pas utile de le modifier ;
- **hw ether** *adresse-mac* permet d'utiliser une autre adresse MAC (Ethernet) que celle de la carte réseau.

> ⓘ En ligne de commandes sous Windows, la configuration d'une interface passe par la commande **netsh interface ip** (voir section 2.2 page 14), bien qu'**ipconfig** (voir section 2.1 p. 12) puisse servir, notamment pour afficher la configuration.

Exemple 1

Affichage de la configuration courante des interfaces actives :

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:21:9b:dd:db:f9
          inet adr:139.124.187.55  Bcast:139.124.187.255  Masque:255.255.255.0
          adr inet6: fe80::221:9bff:fedf:dbf9/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:116984 errors:0 dropped:0 overruns:0 frame:0
          TX packets:48298 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:148500091 (141.6 MiB)  TX bytes:3921033 (3.7 MiB)
          Interruption:17

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:10964 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10964 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          RX bytes:654158 (638.8 KiB)  TX bytes:654158 (638.8 KiB)
```

➪ les interfaces `eth0` et `lo` (*loopback*) sont actives. Pour `eth0`, on peut noter la configuration mise en gras sur l'exemple :

- adresse MAC : `00:21:9b:dd:db:f9`
- adresse IPv4 : `139.124.187.55`
- adresse de diffusion dirigée dans le réseau : `139.124.187.255`
- masque de sous-réseau : `255.255.255.0`
- MTU : `1500`

Affichage de la configuration courante de toutes les interfaces :

```
# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:21:9b:dd:db:f9
          inet adr:139.124.187.55  Bcast:139.124.187.255  Masque:255.255.255.0
          ...
```

```
lo        Link encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          ...

tap426275 Link encap:Ethernet  HWaddr 46:86:d4:80:98:8b
          inet adr:172.23.0.254  Bcast:172.23.255.255  Masque:255.255.255.255
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:1212 errors:0 dropped:0 overruns:0 frame:0
          TX packets:59 errors:0 dropped:34 overruns:0 carrier:0
          collisions:0 lg file transmission:500
          RX bytes:39060 (38.1 KiB)  TX bytes:8629 (8.4 KiB)

wlan0     Link encap:Ethernet  HWaddr 00:1f:3c:c1:05:e6
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:33446 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10817 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:14998208 (14.3 MiB)  TX bytes:4210593 (4.0 MiB)
```

➪ l'interface `wlan0` correspond à la première carte Wifi ; l'interface `tap426275` est une interface virtuelle créée par le simulateur Marionnet pour avoir un accès réseau à une machine virtuelle.

Configuration de l'interface **`eth0`** avec l'adresse `150.151.152.1/16` :

```
# ifconfig eth0 150.151.152.1 netmask 255.255.0.0
```

Vérification :

```
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:21:9b:dd:db:f9
          inet adr:150.151.152.1  Bcast:150.151.255.255  Masque:255.255.0.0
          ...
```

□

Exercice 1 (Consultation et configuration des interfaces)

Nous allons travailler sur des machines virtuelles mises en réseau grâce au simulateur **Marionnet**.

1. Télécharger le fichier `adm_linux.mar` (depuis le site de l'enseignement)
2. Lancer le simulateur **Marionnet** via le menu *Applications* ⟶ *Éducation* ⟶ *Marionnet*
3. Cliquer sur la fenêtre de bienvenue puis charger le fichier `adm_linux.mar` par le menu *Projet* ⟶ *Ouvrir* (prend un temps plus ou moins long)
4. Ce projet contient le réseau `10.0.2.0/24` présenté à la figure 1. Il est formé par :
   - une machine virtuelle linux m1 non configurée ;
   - une machine virtuelle linux m2 déjà configurée avec une adresse dans le réseau `10.0.2.0/24` ;
   - une passerelle vers Internet (de type box des FAI) nommée G1 et d'adresse `10.0.2.2`. Elle est configurée pour donner accès (partiellement) à Internet aux hôtes du réseau. Cette passerelle fait office de routeur par défaut.
   - un switch S1 qui connecte tout ce beau monde.

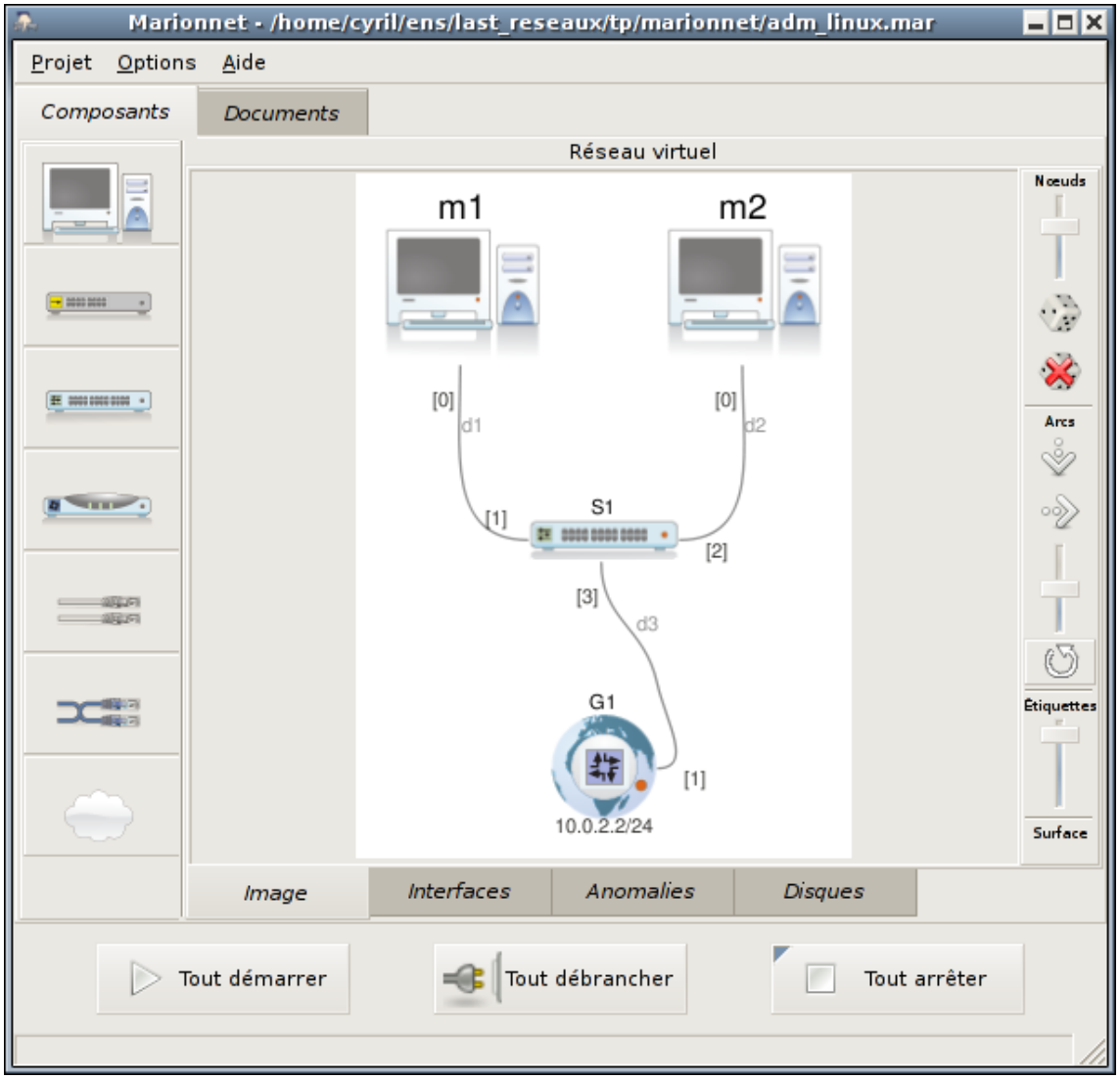

FIGURE 1 – Interface de Marionnet suite au chargement de `adm_linux.mar`

5. Cliquer sur le bouton *Tout Démarrer* en bas à gauche de l'interface. Les machines virtuelles démarrent chacune dans une fenêtre terminal dédiée.
6. Se loger sur m2 en tant que `root` (mot de passe **`root`**)
7. Faire afficher la configuration de ses interfaces. Repérer l'interface Ethernet déjà configurée et déterminer :
   (a) son adresse MAC (Ethernet)
   (b) son adresse IPv4
   (c) l'adresse de diffusion dirigée dans ce réseau
   (d) son masque de sous-réseau
   (e) le MTU de son réseau
8. Se loger sur m1 en tant que `root` (mot de passe **`root`**)
9. Faire afficher la configuration de ses interfaces
10. Configurer son interface **`eth0`** avec l'adresse `10.0.2.15/24`
11. Afficher la configuration de `eth0`

[Corrigé]

1.3 Test de connectivité avec ping

La commande **ping** est disponible sur les systèmes Unix et Windows (voir section 2.3, page 16). Elle permet de tester l'acheminement de datagrammes sur le réseau et, accessoirement, de vérifier qu'une machine est bien présente sur le réseau. Elle permet aussi de réaliser des statistiques sur les temps de réponse ainsi que sur le pourcentage de paquets perdus.

> ⓘ Sous Linux, la commande **ping** se trouve dans le répertoire `/bin`, déjà présent dans le **`PATH`** d'un utilisateur normal.

ping utilise par défaut le protocole ICMP en envoyant des messages (ICMP) de type "Demande d'ECHO" (ECHO REQUEST) demandant au destinataire de répondre par un "Réponse d'ECHO" (ECHO REPLY). Sur certains systèmes, **ping** effectue plusieurs envois puis s'arrête en fournissant des statistiques sur le temps de propagation aller-retour (*Round Trip Time*). Sur d'autres systèmes (comme Linux), il faut arrêter **ping** en tapant `Ctrl-C`.

Ainsi, lorsqu'une réponse arrive, on est assuré que l'ordinateur qu'on utilise est correctement configuré, de même que l'ordinateur intérrogé, que les réseaux qui les séparent sont opérationnels et que les routeurs intermédiaires sont correctement configurés.

Exercice 2 (tests de connectivité depuis m2)

Ouvrir un terminal sur votre PC, et consulter le manuel en ligne de **ping** en tapant :

```
$ man ping
```

pour étudier ses options **-c**, **-b** et **-f**.

Depuis la machine virtuelle m2 :

1. Taper **`ping 10.0.2.2`** pour tester la connectivité avec la passerelle Internet (G1). Vous devriez obtenir les réponses. Taper `Ctrl-C` pour arrêter la commande ;
2. Reprendre la commande précédente et utiliser l'option permettant de limiter le nombre de requêtes à 3 ;
3. Tester la connectivité avec m1 qui devrait avoir l'adresse `10.0.2.15`. Si m1 a été correctement configurée, ses réponses doivent revenir ;
4. Tester la connectivité avec allegro (`139.124.187.4`). Si la passerelle Internet fonctionne normalement, vous devriez encore obtenir des réponses.
5. Tester la connectivité avec l'hôte `192.168.10.30` en limitant à 3 requêtes (il y a peu de chances d'avoir une réponse...) et en utilisant l'option **-f** pour afficher une trace des requêtes transmises (un point par requête).

[Corrigé]

Exercice 3 (tests de connectivité depuis m1)

Depuis m1 :

1. Tester la connectivité avec G1. Vous devriez obtenir les réponses ;
2. Tester la connectivité avec allegro (`139.124.187.4`). Vous devriez obtenir un message d'erreur. Pourtant, aucun matériel (S1, G1, allegro) n'a de configuration particulière excluant m1 de l'accès à Internet. Le problème vient uniquement de m1. D'après vous, quel est-il ?

ⓘ Aides (si besoin) :

- un indice est déjà donné par l'erreur indiquée par **ping** ;
- un autre est qu'allegro n'appartient pas au réseau local de m1 ;
- un dernier est de se demander quels sont les paramètres que nous avons eu besoin d'entrer pour configurer correctement un hôte sur Packet Tracer ? Peut être un élément de configuration manque encore...

[Corrigé]

1.4 Configuration d'une table de routage

Les informations que l'on peut (ou doit) spécifier lorsqu'on ajoute une route dans une table de routage sont :

- le type de destination (réseau ou hôte) ;
- son adresse IP ;
- le masque de cette adresse (afin de prendre en compte des regroupements de réseaux) s'il est différent de celui de la classe du réseau ;
- le routeur associé (`0.0.0.0` si la destination est directement accessible) ;
- l'interface permettant de contacter le routeur (`lo`, `eth0`,...).

La configuration de la table de routage se fait au moyen de la commande **route** sur Unix et Windows (voir section 2.5, page 17). Cette commande permet d'ajouter ou de supprimer des routeurs vers des réseaux ou des stations. Dans notre cas, le synopsis suivant devrait suffire.

Synopsis

route [**-n**]
route add -net *adresse-destination* **netmask** *masque* **gw** *routeur* [**dev** *interface*]
route del -net *adresse-destination* **netmask** *masque* **gw** *routeur* [**dev** *interface*]

La première forme permet d'afficher la table de routage. L'option **-n** désactive la résolution inverse DNS, ce qui est parfois bien pratique. La seconde permet d'ajouter une route passant par le routeur *routeur* vers la destination *adresse-destination* de masque *masque*. Si besoin, l'interface à utiliser pour cette route peut être indiquée avec **dev** *interface*. La dernière forme permet de supprimer une route.

✍ Il n'est pas nécessaire d'ajouter dans la table le réseau auquel est connectée l'interface de la station. En effet, sa configuration avec **ifconfig** a automatiquement provoqué son ajout dans la table.

ⓘ Il est possible avec **route** d'interdire une destination en utilisant l'option **reject** mais cela ne remplace pas l'utilisation d'un bon *firewall*...

Exemple 2

Supposons que nous voulons ajouter une route vers la destination `139.124.110.0/24` qui passe par le routeur `150.151.152.250` et **une route par défaut** qui passe par le routeur `150.151.152.153`.

Dans un premier temps, supposons que l'hôte est déjà configuré au niveau IP, et consultons sa table de routage :

```
# route -n
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
150.151.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth0
```

On ajoute ensuite la route vers `139.124.187.0/24` :

```
# route add -net 139.124.110.0 netmask 255.255.255.0 gw 150.151.152.250
```

puis la route par défaut :

```
# route add -net 0.0.0.0 netmask 0.0.0.0 gw 150.151.152.153
```

Puis on peut vérifier en affichant la table :

```
# route -n
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
150.151.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth0
139.124.110.0   150.151.152.250 255.255.255.0   UG    0      0        0 eth0
0.0.0.0         150.151.152.153 0.0.0.0         UG    0      0        0 eth0
```

Puis en testant la connectivité d'un hôte (dont on sait qu'il répond en temps normal) :

```
# ping 139.124.110.4
PING 139.124.187.4 (139.124.110.4) 56(84) bytes of data.
64 bytes from 139.124.110.4: icmp_seq=1 ttl=64 time=0.276 ms
64 bytes from 139.124.110.4: icmp_seq=2 ttl=64 time=0.485 ms
64 bytes from 139.124.110.4: icmp_seq=3 ttl=64 time=0.408 ms
64 bytes from 139.124.110.4: icmp_seq=4 ttl=64 time=0.363 ms
CTRL+C
--- 139.124.187.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2998ms
rtt min/avg/max/mdev = 0.276/0.383/0.485/0.075 ms
```

□

ⓘ La colonne **Indic** (ou **flags**, selon l'installation) contient une combinaison d'indicateurs donnant quelques renseignements sur la route. Parmi les indicateurs possibles, il y a **U**, **H**, **G**, **D**, **M** et **!** :

- **U** : la route est en service (activée).
- **H** : la destination est un ordinateur (*host*). Sans cet indicateur, la destination est un réseau.
- **G** : la route n'est pas directe et la passerelle est un routeur (*gateway*). Sans cet indicateur, la destination est directement accessible.
- **D** : la route a été créée par une redirection (message ICMP).
- **M** : la route a été modifiée par une redirection (message ICMP).
- **!** : la route est rejetée (option **reject**).

Dans certaines implémentations, la table de routage peut contenir les adresses de la station elle-même. Par exemple, pour la station d'adresse `139.124.187.4`, on pourrait avoir comme table :

```
Destination     Passerelle      Genmask         Indic Metric Ref Use Iface
139.124.187.4   0.0.0.0         255.255.255.255 UH    0      0     0 eth0
139.124.187.0   0.0.0.0         255.255.255.0   U     10     0     0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0     0 lo
0.0.0.0         139.124.187.1   0.0.0.0         UG    10     0     0 eth0
```

Exercice 4 (Consultation de la table de routage de m2)

Sur m2, utiliser **route** avec et sans l'option **-n** pour faire afficher sa table de routage.

[Corrigé]

Exercice 5 (Modification de la table de routage de m1)

Sur m1 :

1. faire afficher sa table de routage ;
2. ajouter une route par défaut passant par la passerelle Internet (`10.0.2.2`) ;
3. faire à nouveau afficher sa table de routage et vérifier cette nouvelle route ;
4. Tester la connectivité d'allegro (`139.124.187.4`). Cette fois, cela devrait fonctionner.

[Corrigé]

✍ Un raccourci pour créer une route par défaut est d'utiliser la forme :

route add default gw *routeur* [**dev** *interface*]

Exercice 6 (routes du PC)

Sur un terminal de votre PC, faire afficher les routes connues. Les routes utilisant les interfaces **tap*** offrent une connectivité avec les machines virtuelles de **Marionnet**. Nous y reviendrons ultérieurement.

[Corrigé]

1.5 Consultation et modification du cache ARP

La commande **arp** permet de consulter et de modifier le cache ARP, sous Unix (Linux) et Windows (voir section 2.4, page 16).

✍ Sous Linux, la commande **arp** est située dans le répertoire `/usr/sbin` qui ne figure pas par défaut dans le **PATH** d'un utilisateur normal.

Exemple 3

Voici un petit extrait des informations que l'on peut obtenir avec **arp** (sur allegro) :

```
$ /usr/sbin/arp
Address                  HWtype  HWaddress           Flags Mask            Iface
b122.iut.univ-aix.fr     ether   f0:4d:a2:24:24:5a   C                     eth0
imp-a.iut.univ-aix.fr            (incomplete)                              eth0
b118.iut.univ-aix.fr     ether   f0:4d:a2:23:6a:e9   C                     eth0
pcdl.iut.univ-aix.fr     ether   b8:ac:6f:30:7d:65   C                     eth0
iutextreme.iut.univ-aix  ether   00:23:89:57:5e:a3   C                     eth0
...
$ /usr/sbin/arp -n
```

```
Address                  HWtype  HWaddress           Flags Mask            Iface
139.124.187.122          ether   f0:4d:a2:24:24:5a   C                     eth0
139.124.187.94                   (incomplete)                              eth0
139.124.187.118          ether   f0:4d:a2:23:6a:e9   C                     eth0
139.124.187.248          ether   b8:ac:6f:30:7d:65   C                     eth0
139.124.187.1            ether   00:23:89:57:5e:a3   C                     eth0
...
```

➪ on voit que l'entrée `139.124.187.94` (`imp-a.iut.univ-aix.fr`) est incomplète car allegro cherche à la résoudre (attend la réponse ARP de sa requête). Les entrées marquées **C** dans la colonne *Flags* sont des entrées temporaires normales.

□

Exercice 7 (manipulation du cache ARP de m1)

Sur un terminal du PC, consulter le manuel en ligne Linux de **arp** en tapant :

```
$ man arp
```

et étudier ses options **-a**, **-n**, **-d** et **-s**.

Puis, sur la machine virtuelle m1 :

1. afficher la liste de toutes les associations présentes dans son cache ARP. Utiliser l'option **-n** pour désactiver la résolution inverse qui n'est pas souhaitée. Selon le temps passé avant cette question, il peut être vide, ce qui est normal car les entrées du cache expirent rapidement (ont une durée de vie très courte).
2. taper **`ping -c 1 10.0.2.2`**
3. afficher la liste de toutes les associations présentes dans le cache ARP. Cette fois nous devrions voir apparaître une entrée pour `10.0.2.2`. Pourquoi ?
4. utilser **arp** pour ajouter manuellement (et de façon permanente) l'entrée `10.0.2.10` associée à l'adresse MAC `00:21:9b:dd:db:f9` puis vérifier le cache
5. taper **`ping -c 1 10.0.2.10`**. Vous devriez ne pas obtenir de réponse. Pouvez-vous expliquer pourquoi ?

   ⓘ Aide (si besoin) :
   - une entrée permanente n'est pas remise en cause
   - quelle est l'adresse MAC de l'interface `eth0` de m2 ?

   ✍ On ne devrait jamais avoir besoin d'ajouter une entrée manuellement dans le cache ARP...

6. supprimer du cache l'entrée pour `10.0.2.10` puis vérifier l'état du cache
7. taper **`ping -c 1 10.0.2.10`**. Cette fois, vous devriez obtenir une réponse. Comprenez-vous pourquoi ?

[Corrigé]

Exercice 8 (petites questions sur arp)

1. Comme nous le verrons plus tard, l'adresse de allegro est `139.124.187.4` et son masque est `255.255.255.0`. Pensez-vous que son cache puisse contenir une association pour l'adresse `139.124.5.51` ?

   ⓘ Aide (si besoin) : ces hôtes se trouvent-ils sur le même réseau ?

2. Trouvez-vous dans le manuel de **arp** une option permettant d'obtenir l'adresse physique d'un hôte (station ou routeur) non présent dans le cache ?

[Corrigé]

1.6 traceroute : connaître le chemin suivi par les datagrammes

La commande **traceroute** permet de connaître le chemin (route) que suivent les datagrammes envoyés vers un hôte donné. Son homologue sous Windows est tracert (voir section 2.6, page 20).

✍ **traceroute** affiche les adresses IP des routeurs traversés pour atteindre l'hôte. Cependant, en cas de routage dynamique, ce ne sera peut être plus le chemin utilisé pour des envois ultérieurs vers ce même hôte.

Elle permet ainsi de savoir à quel endroit bloque la transmission d'un paquet que l'on tente d'envoyer sans succès (malheureusement, ça arrive).

Pour afficher ces routeurs, elle provoque une erreur d'acheminement sur chaque routeur par lequel passe le datagramme IP en agissant sur le champ TTL de ce dernier. En effet, **traceroute** commence par envoyer un datagramme UDP véhiculé par un datagramme IP avec un TTL positionné à 1. Le premier routeur rencontré détruit le datagramme et renvoie une erreur ICMP de TTL expiré. On obtient ainsi l'adresse du premier routeur de la route. **traceroute** envoie ensuite un datagramme UDP dans un datagramme IP avec un TTL à 2 pour connaître le second routeur, et ainsi de suite, jusqu'à atteindre la destination spécifiée (mais sur un port non attribué pour recevoir un message ICMP de port inaccessible).

Exercice 9 (traceroute sur le PC)

ⓘ Sous Linux, la commande **traceroute** se trouve dans le répertoire `/usr/sbin` qui ne figure pas dans le **PATH** d'un utilisateur normal...

Sur un terminal du PC, consulter le manuel en ligne Linux de **traceroute** en tapant :

```
$ man traceroute
```

et étudier ses options **-n** et **-f**.

Sur ce même terminal, utiliser **traceroute** afin de déterminer les routes suivantes :

1. pour atteindre `paprika.iut.univ-aix.fr`
2. pour atteindre `www.free.fr`

> ⓘ Certainement à des fins de confidentialité, certains routeurs ne renvoient pas d'erreur ICMP. Cela se traduit dans **traceroute** par un *timeout* pour l'envoi et l'affichage d'une étoile plutôt que de l'adresse (ou le nom) du routeur. Puisque, pour chaque routeur, **traceroute** fait 3 tentatives, il y aura alors 3 étoiles (3 timeout). **traceroute** est alors considérablement ralenti. Si, à partir d'un certain point, plusieurs routeurs de suite ne répondent pas, c'est probablement que les messages sont filtrés et qu'il n'y aura pas d'espoir d'en savoir davantage. On peut alors arrêter **traceroute**.

3. pour atteindre `saphir.lidil.univ-mrs.fr` en demandant d'afficher les adresses IP des routeurs plutôt que leurs noms.
4. pour atteindre `www.nasa.gov`
5. reprendre la question 2 en demandant à ce que les cinq premiers routeurs n'apparaissent pas (il faut agir sur le TTL du premier datagramme envoyé par **traceroute**)

[Corrigé]

> ⓘ Il existe des versions graphiques de **traceroute**, notamment **xtraceroute** (qui n'est pas installé sur nos stations) mais la localisation géographique des routeurs est loin d'être vraiment précise...

2 Commandes réseau dans l'environnement Windows

Cette section regroupe les équivalents Windows des commandes étudiées. Nous n'avons pas forcément le temps de les étudier en exercices mais il est bon d'avoir un aperçu de leur utilisation et affichage. La section est constituée principalement d'exemples d'utilisation.

> ✍ Les exemples d'exécution des commandes sont tapés sur une fenêtre terminal de Windows XP (émulation MSDOS), lancée via le menu *Démarrer* ⟶ *Exécuter* ⟶ **cmd**.

2.1 ipconfig

ipconfig permet de consulter la configuration IP, et d'obtenir des informations plus détaillées sur les interfaces et autres éléments de configuration. Elle permet de réaliser des opérations élémentaires DHCP comme le renouvellement d'une adresse ou sa résiliation.

Exemple 4

Obtenir de l'aide sur **ipconfig** :

```
C:>ipconfig/?

UTILISATION :
    ipconfig [/? | /all | /renew [carte] | /release [carte] |
              /flushdns | /displaydns | /registerdns |
              /showclassid carte |
              /setclassid carte [ID de classe] ]
...
```

Afficher les informations principales sur la configuration IP :

```
C:\>ipconfig

Configuration IP de Windows


Carte Ethernet Connexion au réseau local:

        Suffixe DNS propre à la connexion :
        Adresse IP. . . . . . . . . . . . : 10.0.2.15
        Masque de sous-réseau . . . . . . : 255.255.255.0
        Passerelle par défaut . . . . . . : 10.0.2.2
```

Afficher une synthèse des configurations réseau de l'hôte :

```
C:\>ipconfig/all

Configuration IP de Windows

        Nom de l'hôte . . . . . . . . . . : vb-xp-iut
        Suffixe DNS principal . . . . . . :
        Type de noud . . . . . . . . . . : Inconnu
        Routage IP activé . . . . . . . . : Non
        Proxy WINS activé . . . . . . . . : Non

Carte Ethernet Connexion au réseau local:

        Suffixe DNS propre à la connexion :
        Description . . . . . . . . . . . : Carte AMD PCNET Family Ethernet PCI
        Adresse physique . . . . . . . . .: 08-00-27-71-76-54
        DHCP activé. . . . . . . . . . . : Oui
        Configuration automatique activée . . . . : Oui
        Adresse IP. . . . . . . . . . . . : 10.0.2.15
        Masque de sous-réseau . . . . . . : 255.255.255.0
        Passerelle par défaut . . . . . . : 10.0.2.2
        Serveur DHCP. . . . . . . . . . . : 10.0.2.2
        Serveurs DNS . . . . . . . . . .  : 212.27.40.240
                                            212.27.40.241
        Bail obtenu . . . . . . . . . . . : jeudi 12 avril 2012 14:33:28
        Bail expirant . . . . . . . . . . : vendredi 13 avril 2012 14:33:28
```

> ✍ Sous Windows, les interfaces sont nommées comme ici "**Connexion au réseau local**". On voit aussi que la représentation d'une adresse Ethernet change sur Windows (`08-00-27-71-76-54`).

Utiliser **ipconfig** pour résilier une allocation DHCP (pour une interface configurée pour DHCP, voir **netsh**) :

```
C:>ipconfig/release

Configuration IP de Windows
```

```
Carte Ethernet Connexion au réseau local:

        Suffixe DNS propre à la connexion :
        Adresse IP. . . . . . . . . . . . : 0.0.0.0
        Masque de sous-réseau . . . . . . : 0.0.0.0
        Passerelle par défaut . . . . . . :
```

et pour renouveler sa configuration par DHCP :

```
C:>ipconfig/renew

Configuration IP de Windows


Carte Ethernet Connexion au réseau local:

        Suffixe DNS propre à la connexion : iut.univ-aix.fr
        Adresse IP. . . . . . . . . . . . : 10.0.2.15
        Masque de sous-réseau . . . . . . : 255.255.255.0
        Passerelle par défaut . . . . . . : 10.0.2.2
```

□

2.2 netsh

La commande **netsh** regroupe un ensemble de modules permettant d'agir sur la configuration réseau d'un poste Windows. Pour configurer les paramètres IPv4 d'une interface réseau, il faut utiliser le module **netsh interface ip**. Il permet notamment d'affecter manuellement (statiquement) les paramètres IPv4 ou de configurer l'interface pour les obtenir par DHCP (dynamiquement).

Exemple 5

Obtenir de l'aide sur **netsh interface ip** :

```
C:>netsh interface ip

Les commandes suivantes sont disponibles :

Commandes dans ce contexte :
?               - Affiche une liste de commandes.
add             - Ajoute une entrée de configuration à une table.
delete          - Supprime une entrée de configuration d'une table.
dump            - Affiche un script de configuration.
help            - Affiche une liste de commandes.
reset           - Réinitialisez TCP/IP et les composants associés.
set             - Définit l'information de configuration.
show            - Affiche les informations.

Pour consulter l'aide d'une commande, entrez la commande, suivie par un espace, et ensu
 entrez ?.
```

Fixer statiquement l'adresse IPv4 `10.0.2.100/24` pour l'interface `Connexion au réseau local`. Fixer aussi les paramètres (optionnels) *Gateway* et *métrique* de la route par défaut à `10.0.2.2` et `20` :

```
C:>netsh int ip set address "Connexion au réseau local" static 10.0.2.100 255.255.255.0

Ok.
```

```
Vérifications de la configuration :
C:>ipconfig

Configuration IP de Windows


Carte Ethernet Connexion au réseau local:

        Suffixe DNS propre à la connexion :
        Adresse IP. . . . . . . . . . . . : 10.0.2.100
        Masque de sous-réseau . . . . . . : 255.255.255.0
        Passerelle par défaut . . . . . . : 10.0.2.2
```

```
C:>route print
===========================================================================
Liste d'Interfaces
0x1 ........................... MS TCP Loopback interface
0x2 ...08 00 27 71 76 54 ...... Carte AMD PCNET Family Ethernet PCI - Miniport d'ordonn
===========================================================================
===========================================================================
Itinéraires actifs :
Destination réseau    Masque réseau  Adr. passerelle   Adr. interface Métrique
          0.0.0.0          0.0.0.0        10.0.2.2      10.0.2.100       20
         10.0.2.0    255.255.255.0      10.0.2.100      10.0.2.100       20
       10.0.2.100  255.255.255.255       127.0.0.1       127.0.0.1       20
   10.255.255.255  255.255.255.255      10.0.2.100      10.0.2.100       20
        127.0.0.0        255.0.0.0       127.0.0.1       127.0.0.1       1
        224.0.0.0        240.0.0.0      10.0.2.100      10.0.2.100       20
  255.255.255.255  255.255.255.255      10.0.2.100      10.0.2.100       1
Passerelle par défaut :          10.0.2.2
===========================================================================
Itinéraires persistants :
  Aucun
```

Configurer la même interface pour obtenir les informations par DHCP (et abandonner la configuration statique) puis vérification :

```
C:>netsh int ip set address "Connexion au réseau local" dhcp
Ok.
```

```
C:>ipconfig

Configuration IP de Windows
```

```
Carte Ethernet Connexion au réseau local:

        Suffixe DNS propre à la connexion : iut.univ-aix.fr
        Adresse IP. . . . . . . . . . . . : 10.0.2.15
        Masque de sous-réseau . . . . . . : 255.255.255.0
        Passerelle par défaut . . . . . . : 10.0.2.2
```

□

2.3 ping

La commande **ping** sous Windows a les mêmes fonctionnalités que sous Unix, bien qu'elle n'admette pas tout à fait les mêmes options. Notamment, elle s'arrête par défaut après l'émission de 4 requêtes.

Exemple 6

Obtenir de l'aide sur **ping** :

```
C:>ping/?

Utilisation : ping [-t] [-a] [-n échos] [-l taille] [-f] [-i vie] [-v TypServ]
                  [-r NbSauts] [-s NbSauts] [[-j ListeHôtes] | [-k ListeHôtes]]
                  [-w Délai] NomCible

Options :
  -t             Envoie la requête ping sur l'hôte spécifié jusqu'à
                 interruption.
                 Entrez Ctrl-Attn pour afficher les statistiques et continuer,
                 Ctrl-C pour arrêter.
  -a             Recherche les noms d'hôte à partir des adresses.
  -n échos       Nombre de requêtes d'écho à envoyer.
```

Tester l'accissibilité d'un hôte :

```
C:>ping 139.124.187.4

Envoi d'une requête 'ping' sur 139.124.187.4 avec 32 octets de données :

Réponse de 139.124.187.4 : octets=32 temps=58 ms TTL=127
Réponse de 139.124.187.4 : octets=32 temps=48 ms TTL=127
Réponse de 139.124.187.4 : octets=32 temps=57 ms TTL=127
Réponse de 139.124.187.4 : octets=32 temps=47 ms TTL=127

Statistiques Ping pour 139.124.187.4:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 47ms, Maximum = 58ms, Moyenne = 52ms
```

□

2.4 arp

La commande **arp** sous Windows a les mêmes fonctionnalités que sous Unix.

Exemple 7

Obtenir de l'aide sur **arp** :

```
C:\>arp/?

Affiche et modifie les tables de traduction d'adresses IP en adresses physiques
utilisées par le protocole de résolution d'adresses ARP.

ARP -s inet_addr eth_addr [if_addr]
ARP -d inet_addr [if_addr]
ARP -a [inet_addr] [-N if_addr]

  -a            Affiche les entrées ARP en cours en interrogeant les données
                en cours du protocole. Si inet_addr est spécifié, seules les
                adresses IP et physiques de l'ordinateur spécifié sont
                affichées. Si plus d'une interface réseau utilise ARP, les
                entrées de chaque table ARP sont affichées.
...

Exemples :
  > arp -s 157.55.85.212   00-aa-00-62-c6-09  .... Ajoute une entrée statique.
  > arp -a                                    .... Affiche la table ARP.

C:\>arp -a

Interface : 10.0.2.15 --- 0x2
  Adresse Internet      Adresse physique      Type
  10.0.2.2              52-54-00-12-35-02     dynamique
```

□

2.5 route

Comme sous Unix, la commande **route** sous Windows permet d'afficher la table de routage et de la modifier. Cependant, on notera que les informations figurant dans les tables Windows sont sensiblement différentes que celles affichées sous Linux.

Exemple 8

Obtenir de l'aide sur **route** :

```
C:>route

Manipule les tables de routage du réseau.

ROUTE [-f] [-p] [cmde [destin]
                  [MASK MasqueRés] [passerelle] [METRIC coût]  [IF interface]

...
  cmde        Spécifie une des quatre commandes suivantes :
                 PRINT     Affiche un itinéraire
                 ADD       Ajoute un itinéraire
```

```
                DELETE    Supprime un itinéraire
                CHANGE    Modifie un itinéraire existant
  destin      Spécifie l'hôte destination.
  MASK        Si le mot clé MASK est présent, le paramètre qui le suit est
              interprété en tant que paramètre de masque réseau.
  MasqueRés   Spécifie la valeur éventuelle du sous-masque réseau à associer
              avec cette entrée d'itinéraire. La valeur par défaut est :
              255.255.255.255.
  passerelle  Spécifie la passerelle.
  interface   numéro d'interface pour l'itinéraire spécifié.
  METRIC      Spécifie le coût métrique pour la destination

...

Exemples :

    > route PRINT
    > route  ADD 157.0.0.0   MASK 255.0.0.0    157.55.80.1 METRIC  3 IF 2

           destination^             ^masque passerelle^    métrique^    ^
                                                              interface^

      Si IF n'est pas fourni, la meilleure interface pour une passerelle
      donnée est recherchée.
    > route PRINT
    > route PRINT 157*   .... N'imprime que les adresses commençant par 157*
    > route CHANGE 157.0.0.0 MASK 255.0.0.0 157.55.80.5 METRIC 2 IF 2

      CHANGE est utilisé pour modifier la passerelle et/ou la métrique seulement
.
    > route PRINT
    > route DELETE 157.0.0.0
    > route PRINT
```

Supposons que la configuration IP actuelle d'un hôte est la suivante :

```
C:>ipconfig

Configuration IP de Windows


Carte Ethernet Connexion au réseau local:

        Suffixe DNS propre à la connexion :
        Adresse IP. . . . . . . . . . . . : 10.0.2.15
        Masque de sous-réseau . . . . . . : 255.255.255.0
        Passerelle par défaut . . . . . . : 10.0.2.2
```

Étudions l'affichage de la table de routage :

```
C:>route print
===========================================================================
Liste d'Interfaces
0x1 ........................... MS TCP Loopback interface
```

```
0x2 ...08 00 27 71 76 54 ...... Carte AMD PCNET Family Ethernet PCI - Mini...
===========================================================================
===========================================================================
Itinéraires actifs :
Destination réseau    Masque réseau  Adr. passerelle   Adr. interface Métrique
          0.0.0.0          0.0.0.0        10.0.2.2       10.0.2.15       20
         10.0.2.0    255.255.255.0       10.0.2.15       10.0.2.15       20
        10.0.2.15  255.255.255.255       127.0.0.1       127.0.0.1       20
       10.0.2.255  255.255.255.255       10.0.2.15       10.0.2.15       20
        127.0.0.0        255.0.0.0       127.0.0.1       127.0.0.1       1
        224.0.0.0        240.0.0.0       10.0.2.15       10.0.2.15       20
  255.255.255.255  255.255.255.255       10.0.2.15       10.0.2.15       1
Passerelle par défaut :          10.0.2.2
===========================================================================
Itinéraires persistants :
  Aucun
```

➪ La première partie de la commande liste les interfaces réseau de l'hôte. Windows attribue un numéro à chaque interface. Ce numéro est requis pour l'ajout (**ADD**) et la modification (**CHANGE**) d'une route avec **route**. Dans cet exemple, il y a 2 interfaces :

- l'interface **1** (0x1) est l'interface *loopback* ;
- l'interface **2** (0x2) correspond à la carte réseau de l'hôte. Elle a pour adresse MAC 08-00-27-71-76-54 (selon l'écriture Windows).

On pourra faire référence à une interface en utilisant son numéro en hexadécimal (en laissant le 0x) ou en décimal.
Ensuite, **route** affiche la table de routage qui contient beaucoup plus de lignes que sous Linux (ou que sur un routeur CISCO) sans pour autant ajouter de fonctionnalité supplémentaire. Les routes importantes ont été mises en évidence en gras dans l'exemple :

- la 1re ligne correspond à la route par défaut qui passe par le routeur (passerelle) 10.0.2.2 ;

  ✍ on peut remarquer que la colonne *Adr. interface* indique l'adresse IP de l'interface à utiliser pour la route.

- la 2e ligne correspond à la route directe vers le réseau local ;

  ✍ on peut remarquer qu'au lieu de l'adresse 0.0.0.0 comme passerelle pour les destinations directement accessibles (comme sous Linux), Windows affiche l'adresse IP de l'hôte sur ce réseau.

- la 3e ligne correspond à l'adresse IP de l'hôte. Elle indique que si l'hôte doit discuter avec lui-même, il doit utiliser l'interface *loopback* ;
- la 4e ligne correspond à l'adresse de diffusion dirigée dans le réseau local ;
- la 5e ligne correspond à l'interface *loopback* et regroupe toutes les adresses commençant par 127 ;
- la 6e ligne correspond à une route pour les adresses de classe D (multidiffusion). Ce type d'adresse est aussi compris par Linux mais il ne les fait pas apparaître dans la table de routage ;
- la dernière route correspond à l'adresse de diffusion limitée.

> ✍ On peut remarquer que Linux n'afficherait que les 2 premières routes tout en gérant les adresses formant les autres routes affichées par Windows.

Modification de la route par défaut pour passer par le routeur `10.0.2.1` via l'interface 2 :

```
C:>route CHANGE 0.0.0.0 MASK 0.0.0.0 10.0.2.1 METRIC 2 IF 2
```

Vérification de la modification :

```
C:>route print
===========================================================================
Liste d'Interfaces
0x1 ........................... MS TCP Loopback interface
0x2 ...08 00 27 71 76 54 ...... Carte AMD PCNET Family Ethernet PCI - Mini...
===========================================================================
===========================================================================
Itinéraires actifs :
Destination réseau    Masque réseau  Adr. passerelle   Adr. interface Métrique
          0.0.0.0          0.0.0.0        10.0.2.1       10.0.2.15       2
         10.0.2.0    255.255.255.0        10.0.2.15       10.0.2.15       20
        10.0.2.15  255.255.255.255        127.0.0.1       127.0.0.1       20
   10.255.255.255  255.255.255.255        10.0.2.15       10.0.2.15       20
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1       1
        224.0.0.0        240.0.0.0        10.0.2.15       10.0.2.15       20
  255.255.255.255  255.255.255.255        10.0.2.15       10.0.2.15       1
Passerelle par défaut :          10.0.2.1
===========================================================================
Itinéraires persistants :
  Aucun
```

□

2.6 tracert

La commande **tracert** offre la même fonctionnalité principale que **traceroute** sous Linux, mais admet un nombre réduit d'options.

Exemple 9

Obtenir de l'aide sur **tracert** :

```
C:>tracert

Utilisation : tracert [-d] [-h SautsMaxi] [-j ListeHôtes] [-w délai] NomCible

Options :
   -d                 Ne pas convertir les adresses en noms d'hôtes.
   -h SautsMaxi       Nombre maximum de sauts pour rechercher la cible.
   -j ListeHôtes      Itinéraire source libre parmi la liste des hôtes.
   -w délai           Attente d'un délai en millisecondes pour chaque réponse.
```

Tracer la route vers `www.univmed.fr` en désactivant la résolution inverse :

```
C:>tracert -d www.univmed.fr

Détermination de l'itinéraire vers mozart.pg.univmed.fr [139.124.196.119]
avec un maximum de 30 sauts :

  1    <1 ms     1 ms    <1 ms  10.0.2.2
  2     *         *        *     Délai d'attente de la demande dépassé.
  3    <1 ms    <1 ms    <1 ms  193.50.131.177
  4     5 ms     1 ms     2 ms  192.168.100.33
  5     2 ms     1 ms     1 ms  193.50.131.18
  6     3 ms     2 ms     2 ms  193.50.131.25
  7     3 ms     2 ms     1 ms  139.124.196.119

Itinéraire déterminé.
```

Tracer la route vers `www.google.fr` en limitant à 20 sauts :

```
C:>tracert -h 20 www.google.fr

Détermination de l'itinéraire vers www-cctld.l.google.com [74.125.230.216]
avec un maximum de 20 sauts :

  1    <1 ms    <1 ms    <1 ms  10.0.2.2
  2     *         *        *     Délai d'attente de la demande dépassé.
  3     *         *        *     Délai d'attente de la demande dépassé.
  4     2 ms    <1 ms    <1 ms  193.50.131.225
  5     2 ms     4 ms     2 ms  192.168.100.33
  6    37 ms     2 ms     2 ms  192.168.100.2
  7    15 ms    43 ms    50 ms  vl10-gi8-2-marseille1-rtr-021.noc.renater.fr [193.51.18
  8    12 ms     9 ms    14 ms  te1-1-marseille2-rtr-021.noc.renater.fr [193.51.179.158
  9    11 ms    10 ms    11 ms  193.51.179.182
 10    14 ms    10 ms    19 ms  te0-0-0-1-paris2-rtr-001.noc.renater.fr [193.51.189.2]
 11    21 ms    13 ms    10 ms  te1-1-paris2-rtr-021.noc.renater.fr [193.51.189.9]
 12    10 ms    13 ms    11 ms  193.51.182.197
 13    23 ms    14 ms    17 ms  72.14.238.228
 14    17 ms    12 ms    13 ms  209.85.242.49
 15     8 ms    18 ms    12 ms  par08s09-in-f24.1e100.net [74.125.230.216]

Itinéraire déterminé.
```

□

3 Synthèse de l'utilisation des commandes réseau

Exercice 10 (Déduction de la topologie)

On exécute des commandes sur trois machines A, B et C (stations ou routeurs) d'un ou plusieurs réseaux physiques, dont voici les résultats :

Machine A : (sous Windows)

```
C:>route print
===========================================================================
Liste d'Interfaces
0x1 ........................... MS TCP Loopback interface
0x2 ...00 07 e9 83 0f 6b ...... Intel(R) PRO/100 VE Network Connection -
                                      Miniport d'ordonnancement de paquets
===========================================================================
===========================================================================
Itinéraires actifs :
Destination réseau    Masque réseau  Adr. passerelle   Adr. interface Métrique
          0.0.0.0          0.0.0.0   130.26.144.245     130.26.148.10       20
        127.0.0.0        255.0.0.0        127.0.0.1         127.0.0.1       1
     130.26.144.0    255.255.240.0    130.26.148.10     130.26.148.10       20
      130.26.64.0    255.255.240.0    130.26.149.20     130.26.148.10       20
      130.26.80.0    255.255.240.0    130.26.149.20     130.26.148.10       20
    130.26.148.10  255.255.255.255        127.0.0.1         127.0.0.1       20
   130.26.159.255  255.255.255.255    130.26.148.10     130.26.148.10       20
        224.0.0.0        240.0.0.0    130.26.148.10     130.26.148.10       20
  255.255.255.255  255.255.255.255    130.26.148.10     130.26.148.10       1
Passerelle par défaut :    130.26.144.245
===========================================================================
Itinéraires persistants :
  Aucun
C:>tracert 130.26.80.200

Détermination de l'itinéraire vers 130.26.80.200 avec un maximum de 30 sauts :
  1    <1 ms    <1 ms    <1 ms  130.26.149.20
  2     1 ms     1 ms     1 ms  130.26.68.250
  3     1 ms     1 ms     1 ms  130.26.80.200

Itinéraire déterminé.
C:>ping 130.26.64.1

Envoi d'une requête 'ping' sur 130.26.64.1 avec 32 octets de données :
Réponse de 130.26.64.1 : octets=32 temps<1ms TTL=64
Réponse de 130.26.64.1 : octets=32 temps<1ms TTL=64
Réponse de 130.26.64.1 : octets=32 temps<1ms TTL=64
Réponse de 130.26.64.1 : octets=32 temps<1ms TTL=64

Statistiques Ping pour 130.26.64.1:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms
```

Machine B :

```
$ ifconfig -a
eth0  Lien encap:Ethernet  HWaddr 00:90:27:72:3B:E5
      inet adr:130.26.149.20  Bcast:130.26.159.255 Masque:255.255.240.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      Paquets Reçus:481149199 erreurs:0 jetés:0 débordements:0 trames:0
      Paquets transmis:501617823 erreurs:0 jetés:0 débordements:0 carrier:0
```

```
      collisions:0 lg file transmission:100
      Interruption:19 Adresse de base:0x4000

eth1  Lien encap:Ethernet  HWaddr 00:80:55:72:34:6E
      inet adr:130.26.78.20  Bcast:130.26.79.255  Masque:255.255.240.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      Paquets Reçus:4899 erreurs:0 jetés:0 débordements:0 trames:0
      Paquets transmis:5023 erreurs:0 jetés:0 débordements:0 carrier:0
      collisions:0 lg file transmission:100
      Interruption:19 Adresse de base:0x4000

lo    Lien encap:Boucle locale
      inet adr:127.0.0.1  Masque:255.0.0.0
      UP LOOPBACK RUNNING  MTU:3924  Metric:1
      Paquets Reçus:59841291 erreurs:0 jetés:0 débordements:0 trames:0
      Paquets transmis:59841291 erreurs:0 jetés:0 débordements:0 carrier:0
      collisions:0 lg file transmission:0
$ route -n
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref Use Iface
130.26.144.0    0.0.0.0         255.255.240.0   U     0      0   0   eth0
130.26.64.0     0.0.0.0         255.255.240.0   U     0      0   0   eth1
130.26.80.0     130.26.68.250   255.255.240.0   UG    0      0   0   eth1
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0   0   lo
0.0.0.0         130.26.144.245  0.0.0.0         UG    0      0   0   eth0
```

Machine C :

```
$ ifconfig
eth0  Lien encap:Ethernet  HWaddr 00:0F:1F:10:11:12
      inet adr:130.26.80.200  Bcast:130.26.95.255  Masque:255.255.240.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      Paquets Reçus:12345 erreurs:0 jetés:0 débordements:0 trames:0
      Paquets transmis:6543 erreurs:0 jetés:0 débordements:0 carrier:0
      collisions:0 lg file transmission:100
      Interruption:19 Adresse de base:0x4000

lo    Lien encap:Boucle locale
      inet adr:127.0.0.1  Masque:255.0.0.0
      UP LOOPBACK RUNNING  MTU:3924  Metric:1
      Paquets Reçus:123456 erreurs:0 jetés:0 débordements:0 trames:0
      Paquets transmis:123456 erreurs:0 jetés:0 débordements:0 carrier:0
      collisions:0 lg file transmission:0
$ route -n
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
130.26.80.0     0.0.0.0         255.255.240.0   U     0      0        0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        0 lo
0.0.0.0         130.26.86.110   0.0.0.0         UG    0      0        0 eth0
$ traceroute 130.26.64.1
traceroute to 130.26.64.1 (130.26.64.1), 30 hops max, 40 byte packets
 1  130.26.86.110 (130.26.86.110)  0.656 ms  0.986 ms  1.387 ms
```

```
 2  130.26.64.1 (130.26.64.1)  1.847 ms  2.572 ms  3.298 ms
```

Travail à faire :

Faire le schéma de cette interconnexion de réseaux en précisant tous les équipements (stations et routeurs) déductibles de ces commandes, les interfaces, les adresses IP attribuées à ces interfaces, les noms de machines, si possible leurs OS, ainsi que les adresses réseaux. Le cas échéant, faire ressortir le routeur menant à Internet.

> ⓘ Afin de vous aider, vous pouvez utiliser la table de conversion décimal-binaire de toutes les valeurs possibles sur un octet disponible dans le fichier conversion_decimal_binaire.pdf (lien sur le site).

[Corrigé]

4 Noms de stations et de domaine

Les adresses IP ne sont pas forcément très lisibles et sont difficiles à retenir. C'est pourquoi, il est possible d'associer à une station, un ensemble de noms qui sont plus faciles à retenir. **Cette association peut être officieuse ou officielle.** Si elle est officieuse, elle ne sera reconnue que par les stations qui ont été configurées pour cela. C'est une configuration **locale**. Si elle est officielle, elle sera reconnue par toute station pouvant faire appel à un serveur de noms (quasiment toutes les machines connectées à Internet). Mais cela nécessite une déclaration au DNS (*Domain Name Service*).

Les configurations officielle et officieuse diffèrent quelque peu dans la flexibilité du nommage. Si officieusement, on peut se laisser aller à quelques fantaisies, il n'en est pas de même officiellement.

4.1 Noms officieux

4.1.1 Sous Unix

Les noms officieux sont simplement renseignés sous Unix dans le fichier `/etc/hosts` pour les noms de stations et dans le fichier `/etc/networks` pour les noms de réseaux. C'est une **configuration locale** : les noms déclarés dans ces fichiers ne sont connus que de la station qui les héberge. Lorsque le DNS n'existait pas encore, c'était la seule façon d'utiliser un nom d'hôte plutôt que son adresse.

Ces fichiers sont constitués de lignes commençant par une adresse IP suivie d'un ou plusieurs noms. Toutes les commandes réseaux consultent ces fichiers et remplacent lorsqu'il y a lieu les noms qui leur sont passés en arguments par les adresses correspondantes.

Le fichier `/etc/hosts` peut être modifié en utilisant la commande **hostname** (voir noms officiels).

Exercice 11 (noms officieux sur allegro)

1. Depuis un terminal sur le PC, se connecter sur allegro par **ssh** (**`ssh allegro`**) ;
2. Quels sont les noms d'hôtes officieusement reconnus par allegro ?
   Pour le savoir, consulter simplement le fichier `/etc/hosts`, où le caractère **#** marque le début d'un commentaire.

3. Quels sont les noms de réseaux officieusement reconnus par allegro ?
   Pour le savoir, consulter `/etc/networks`. S'il n'existe pas, c'est qu'il n'y en a pas.

[Corrigé]

Exercice 12 (noms officieux sur m1)

La machine virtuelle m1 n'est pas (encore) configurée au niveau (client) DNS. On ne peut donc pas utiliser des noms de machines. Pour le moment, nous allons procéder à une configuration locale.
Depuis m1 :

1. Taper **`ping -c 2 allegro`**. Cela devrait échouer car le nom allegro est inconnu.
2. Essayer avec **`ping -c 2 allegro.iut.univ-aix.fr`**. On ne devrait pas avoir plus de succès.
3. Éditer le fichier `/etc/hosts` avec **vi** (si, si !) et ajouter une ligne pour associer les noms allegro et allegro.iut.univ-aix.fr à l'adresse `139.124.187.4` puis sauver le fichier en quittant
4. Retenter les **ping** précédents. Cette fois cela doit fonctionner car la résolution locale traduit les noms en l'adresse `139.124.187.4`

[Corrigé]

4.1.2 Sous Windows

Windows a adopté l'utilisation de noms officieux à la manière d'Unix, à tel point que les fichiers utilisés portent le même nom. Ces fichiers sont `C:\WINDOWS\system32\drivers\etc\hosts` (ou sur Windows 2000, `C:\WINNT\system32\drivers\etc\hosts`) et `C:\WINDOWS\system32\drivers\etc\networks` (ou sur Windows 2000, `C:\WINNT\system32\drivers\etc\networks`).

Exemple 10

Sur la ligne de commande, on peut faire afficher ces fichiers en utilisant la commande **type** (sorte de **cat**).
Sous Windows XP, cela donne :

```
C:>type c:\windows\system32\drivers\etc\hosts
# Copyright (c) 1993-1999 Microsoft Corp.
#
# Ceci est un exemple de fichier HOSTS utilisÚ par Microsoft TCP/IP
# pour Windows.
#
# Ce fichier contient les correspondances des adresses IP aux noms d'hôtes.
# Chaque entrÚe doit Ûtre sur une ligne propre. L'adresse IP doit Ûtre placÚe
# dans la premiÞre colonne, suivie par le nom d'hôte correspondant. L'adresse
# IP et le nom d'hôte doivent Ûtre sÚparÚs par au moins un espace.
#
# De plus, des commentaires (tels que celui-ci) peuvent Ûtre insÚrÚs sur des
# lignes propres ou aprÞs le nom d'ordinateur. Ils sont indiquÚ par le
# symbole '#'.
#
# Par exemple :
#
#      102.54.94.97     rhino.acme.com          # serveur source
#       38.25.63.10     x.acme.com              # hôte client x
```

```
127.0.0.1       localhost

C:>type c:\windows\system32\drivers\etc\networks
# Copyright (c) 1993-1999 Microsoft Corp.
#
# Ce fichier contient les correspondances des noms et des numéros de réseau
# pour  les réseaux locaux. Les numéros de réseau sont reconnus en forme
# décimale séparée par des points.
#
# Format:
#
# <nom réseau>  <numéro réseau>     [alias...]  [#<commentaires>]
#
# Par exemple:
#
#    loopback     127
#    campus       284.122.107
#    londres      284.122.108

loopback                127
```

□

4.2 Noms officiels : utilisation et configuration du client DNS

Le format du nom officiel d'un hôte ressemble à *hôte*`.`*domaine*. Dans cette écriture, *domaine* ne correspond pas forcément à un réseau. C'est en fait un **nom de domaine** qui est géré par une organisation. Elle est chargée de mettre à disposition un serveur de noms qui doit répondre aux requêtes provenant du réseau et concernant des caractéristiques de ce domaine : adresse IP d'un hôte de ce domaine, adresse IP du serveur de mail de ce domaine, adresse électronique du responsable de ce domaine...

Le nom d'un domaine est composé de chaînes de caractères, appelées *labels*, séparées par des points. Par exemple, **`iut.univ-aix.fr`** est le domaine du réseau de l'IUT. Le nom officiel —ou **nom complètement qualifié** (**FQDN** pour *Fully Qualified Domain Name*— de allegro est **`allegro.iut.univ-aix.fr.`** (avec un point à la fin).

4.2.1 Commande hostname

Les noms officiels (ou officieux) affectés à un hôte sont consultables avec la commande **hostname** sous Linux et Windows.

> ✍ Sous Linux, la commande **hostname** se trouve dans le répertoire `/bin`

Exercice 13 (hostname sur allegro)

Sur allegro, consulter le manuel en ligne de **hostname** en tapant :

```
$ man hostname
```

et étudier ses options **-f**, **-s** et **-d**.

Toujours sur allegro, utiliser **hostname** pour afficher :

1. le nom complet d'allegro
2. le nom court d'allegro
3. le nom de domaine d'allegro

[Corrigé]

Commande hostname sur Windows

Sur Windows, **hostname** ne reconnaît pas d'option. On ne peut qu'afficher le nom de l'hôte avec cette commande.

Exemple 11

```
C:>hostname
vb-xp-iut
```

□

4.2.2 Résolution de nom sous Unix et le fichier nsswitch.conf

Sous Unix, lorsqu'un nom d'hôte est utilisé, les commandes réseau utilisent généralement d'abord le fichier **/etc/hosts** avant d'effectuer une requête DNS. D'autres méthodes pour "résoudre" des noms d'hôtes peuvent être aussi utilisées (noms *netbios* ou autres systèmes de nommage). C'est le fichier **/etc/nsswitch.conf** qui indique dans quel ordre la résolution doit être faite.

> ⓘ Le fichier `/etc/nsswitch.conf` provient de Sun Microsystems et de son système Solaris 2 et veut dire "aiguilleur de service de nom" (*name service switch*). Il a été adopté par la majorité des systèmes Unix, il y a déjà un moment.

Il ne sert pas qu'à la résolution de noms, car de nombreuses autres ressources du système peuvent être décentralisées (utilisateurs et mots de passe, groupes d'utilisateurs, alias de messagerie, etc.). Pour chaque ressource, `/etc/nsswitch.conf` contient une ligne qui indique comment la rechercher, par ordre de méthode de recherche.

Pour une *ressource* donnée, la ligne a la simple syntaxe suivante :

ressource : { *service* }

où *service* est le service à utiliser. Lorsqu'il y a plusieurs services, ils sont consultés de gauche à droite jusqu'à trouver une correspondance. Parmi les services possibles il y a :

- **files** : rechercher dans les fichiers locaux. Le(s) fichier(s) à consulter dépend(ent) de la ressource (`/etc/hosts` pour un nom d'hôte, `/etc/passwd` pour un utilisateur, etc.)
- **dns** : rechercher par le service DNS
- **nis** : rechercher par le service NIS (de Sun Microsystems), appelé aussi pages-jaunes (*yellow pages*)
- **ldap** : rechercher dans un annuaire par le protocole LDAP
- **winbind** : rechercher par un serveur Windows (informations utilisateur/groupe, authentification)
- **wins** : rechercher par le service de nommage Windows

- etc.

✍ La *ressource* qui concerne les noms d'hôtes est **`hosts`**. En principe, pour la résolution de noms d'hôtes, c'est le fichier local `/etc/hosts` qui est préféré avant tout autre service (notamment le DNS). Autrement dit, la résolution d'un nom ne se fera pas par le DNS si il figure dans `/etc/hosts`.

Exemple 12

Sur la machine virtuelle m1, le fichier `/etc/nsswitch.conf` contient :

```
passwd:         compat
group:          compat
shadow:         compat

hosts:          files dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

➪ la ligne mise en gras indique que pour la résolution de noms (`hosts`), il faut d'abord utiliser le fichier **`/etc/hosts`** (`files`) et si le nom n'y figure pas, utiliser le DNS (`dns`).

□

Exercice 14 (consultation de /etc/nsswitch.conf sur allegro)

Consulter le fichier `/etc/nsswitch.conf` de allegro pour savoir comment celui-ci procède pour la résolution de noms d'hôtes.

ⓘ Pour en savoir plus sur le fichier `nsswitch.conf`, consulter le manuel de ce fichier en tapant :

```
$ man nsswitch.conf
```

[Corrigé]

4.2.3 Configuration du client (solveur) DNS

Sous Unix, afin d'interroger le DNS pour reconnaître les noms officiels de toutes les stations déclarées, les **solveurs de noms** (clients DNS) se basent sur le contenu du fichier **`/etc/resolv.conf`**, qu'il faut renseigner.

Ce fichier comprend un certain nombre de paramètres indiqués par des mots clés suivis d'informations. Les paramètres principaux sont :

- **domain** *domaine-local*
  pour préciser le domaine de la machine. La recherche d'un nom court d'hôte (i.e. qui ne comporte pas de .) se fera en priorité dans ce domaine. Si ce paramètre n'existe pas, il sera déduit du résultat de **hostname** ;
- **search** *domaine* { *domaine* }
  pour préciser une liste ordonnée de domaines dans lesquels la recherche d'un nom court d'hôte se fera. Ce paramètre est souvent utilisé à la place de **domain** en particulier quand la station n'appartient pas à un domaine particulier ;
- **nameserver** *adresse-ip*
  pour indiquer un serveur de noms DNS à contacter pour la résolution. Plusieurs lignes **nameserver** peuvent être indiquées. Leur ordre d'apparition détermine la priorité du serveur : la première désigne le serveur préféré. Celui qui suit n'est utilisé que si le premier ne répond pas, etc. Il est conseillé de renseigner au moins 2 serveurs de noms, voire 3.

Pour effectuer une requête DNS, comme résoudre un nom en adresse IPv4, le solveur contacte le serveur de noms afin de lui demander l'adresse IP associée au nom recherché. Si le serveur ne connaît pas cette association, il demandera à un autre serveur ce renseignement. C'est le mode standard de **recherche récursive**.

Exercice 15 (consultation de /etc/resolv.conf du PC)

Sur un terminal de votre PC, consulter le fichier `/etc/resolv.conf`, où le caractère **#** commence un commentaire.
Puis, répondre aux questions suivantes :

1. Dans quel(s) domaine(s) par défaut les noms courts d'hôtes sont-ils recherchés par allegro ?
2. Quels sont les serveurs de noms utilisés par votre PC ?

> ⓘ Pour en savoir plus sur le fichier `resolv.conf`, consulter le manuel de ce fichier en tapant :
>
> ```
> $ man resolv.conf
> ```

[Corrigé]

Exercice 16 (configuration du client DNS de m1)

> ⓘ Bien que désormais la modification du fichier `/etc/resolv.conf` devrait passer par des outils appropriés, tels **resolvconf**, nous allons modifier manuellement celui de m1.

Sur la machine virtuelle m1 :

1. Éditer le fichier `/etc/hosts` avec **vi** et supprimer les informations concernant allegro
2. Vérifier que **ping -c 2 allegro** provoque une erreur
3. Éditer le fichier `/etc/resolv.conf` avec **vi** et indiquer les mêmes informations que sur le PC puis sauver le fichier en quittant
4. Si la configuration est correcte, la commande **ping -c 2 allegro** devrait réussir, sinon revoir le contenu du fichier.

[Corrigé]

4.2.4 Interrogation du DNS avec host, dig et nslookup

Certaines commandes permettent d'interroger des serveurs DNS. Il s'agit notamment de **host**, **dig** et de **nslookup**. Ces commandes ne sont pas toujours disponibles sur Windows. Elles admettent un bon nombre de paramétrages, notamment le serveur de nom à utiliser pour l'interrogation. Historiquement, **nslookup** était la plus utilisée. Elle a surtout l'avantage d'être interactive.

La commande host

host est une commande non interactive qui s'utilise en ligne de commandes.

> ✍ Sous Linux **host** se trouve dans le répertoire `/usr/bin`.

Son synopsis le plus basique est le suivant.

Synopsis

host [**-r**] [**-t** *type*] *nom* [*serveur*]

où :

- *nom* est le nom que l'on veut résoudre. Ce peut être une adresse IP, dans ce cas, **host** effectue une **résolution inverse**
- *serveur* est le serveur de noms qui sera utilisé pour la résolution. S'il n'est pas spécifié, le serveur utilisé est celui par défaut (le premier **`nameserver`** de `/etc/resolv.conf`)
- **-r** désactive la recherche récursive. La recherche récursive est activée par défaut et demande au serveur de noms, si celui-ci ne sait pas résoudre le *nom* (pour le *type* de question posée), de contacter un serveur adéquat qui saura le faire (ou qui contactera lui-même un autre serveur, etc.), et d'afficher la réponse
- **-t** *type* précise le type d'information (voir encadré ci-dessous) que l'on souhaite obtenir, correspondant à un ou plusieurs **enregistrements DNS**.

> ✍ Au cours de ce TP, les informations qui nous intéressent sont demandées par les *type*s suivants (la casse importe peu) :
>
> - **`A`** : adresse IPv4 correspondant à *nom* (il existe aussi le type **`AAAA`** pour les adresses IPv6)
> - **`NS`** : (Name Server) serveur de domaines ayant autorité sur le domaine *nom*. Il faut dans ce cas que *nom* soit un domaine (pas un hôte)
> - **`MX`** : (Mail eXchanger) serveur SMTP du domaine *nom*. Il faut dans ce cas que *nom* soit un domaine (pas un hôte)
> - **`PTR`** : enregistrement servant à la **résolution inverse** (d'adresse IPv4 en nom d'hôte)
> - **`ANY`** : tout type d'information disponible pour ce *nom*

Exemple 13

Obtenir l'adresse IPv4 d'allegro.iut.univ-aix.fr :

```
$ host -t a allegro.iut.univ-aix.fr
allegro.iut.univ-aix.fr has address 139.124.187.4
```

Demander la résolution inverse requiert de poser la question en présentant l'adresse `139.124.187.4` dans le domaine **`in-addr.arpa`** :

```
$ host -t ptr 4.187.124.139.in-addr.arpa
4.187.124.139.in-addr.arpa domain name pointer allegro.iut.univ-aix.fr.
```

Obtenir le(s) serveur(s) de noms du domaine univmed.fr :

```
$ host -t ns univmed.fr
univmed.fr name server ns1.univmed.fr.
univmed.fr name server cnudns.cines.fr.
univmed.fr name server dns.irisa.fr.
univmed.fr name server ns2.univmed.fr.
```

Obtenir le(s) serveur(s) SMTP en charge de la réception des courriels envoyés aux utilisateurs du domaine hotmail.fr (adresses électroniques de type `toto@hotmail.fr`) :

```
$ host -t mx hotmail.fr
hotmail.fr mail is handled by 5 mx1.hotmail.com.
hotmail.fr mail is handled by 5 mx2.hotmail.com.
hotmail.fr mail is handled by 5 mx3.hotmail.com.
hotmail.fr mail is handled by 5 mx4.hotmail.com.
```

Obtenir tout type d'information disponible sur le domaine univmed.fr :

```
$ host -t any univmed.fr
univmed.fr mail is handled by 0 mx1.univmed.fr.
univmed.fr mail is handled by 0 mx0.univmed.fr.
univmed.fr name server ns1.univmed.fr.
univmed.fr name server ns2.univmed.fr.
univmed.fr name server cnudns.cines.fr.
univmed.fr name server dns.irisa.fr.
```

□

Exercice 17 (interrogations DNS avec host)

Sur un terminal de votre PC, afficher le manuel en ligne de **host** en tapant :

```
$ man host
```

afin de s'y référer éventuellement.

Depuis un autre terminal du PC, utiliser **host** pour :

1. Obtenir l'adresse IPv4 de l'hôte `www.lsis.org`
2. Obtenir le(s) serveur(s) de noms du domaine `lsis.org` puis l'adresse IPv4 de l'un de ces serveurs si elle n'apparaît pas

[Corrigé]

La commande dig

dig est une commande non interactive qui s'utilise en ligne de commandes.

> ✍ Sous Linux **dig** se trouve dans le répertoire `/usr/bin`.

Son synopsis le plus basique est le suivant.

Synopsis

dig [@*serveur*] *nom type*

où :

- *serveur* est le serveur de noms à contacter
- *nom* est le nom que l'on veut résoudre
- *type* est le type de question posée (comme pour **host**)

Exercice 18 (interrogations DNS avec dig)

Sur un terminal de votre PC, afficher le manuel en ligne de **dig** en tapant :

```
$ man dig
```

afin de s'y référer éventuellement.

Depuis un autre terminal du PC, utiliser **dig** pour :

1. Obtenir le(s) serveur(s) SMTP en charge du domaine `univmed.fr`
2. Obtenir tout type d'information disponible sur www.enseignementsup-recherche.gouv.fr

[Corrigé]

La commande nslookup

À la différence des deux commandes précédentes, **nslookup** est une commande interactive qui offre un invite de commande (*prompt*).

> ✍ Sous Linux **nslookup** se trouve dans le répertoire `/usr/bin`.

Son synopsis le plus basique est le suivant.

Synopsis

nslookup

En l'exécutant, un prompt s'affiche et l'on peut paramétrer le solveur de nom avec les commandes suivantes :

- **set** [**no**]**rec** pour désactiver (**norec**) ou activer (**rec**) le mode récursif. Il est activé par défaut
- **server** *serveur* pour indiquer que l'on veut interroger le serveur de noms *serveur*. Sans cette commande, le serveur utilisé est celui par défaut de l'ordinateur (voir `/etc/resolv.conf`)
- **set q=***type* pour préciser le *type* de question posée. Le *type* par défaut est **A**
- *nom* provoque l'émission d'une requête DNS au serveur choisi précédemment. Elle demande les informations correspondant au *type* choisi et concernant *nom*. Si *nom* ne se termine pas par un point, il n'est pas complètement qualifié (FQDN) et si la recherche échoue pour ce *nom*, des requêtes supplémentaires seront émises en ajoutant à *nom* les domaines de recherche de la machine (lignes **domain** et **search** dans `/etc/resolv.conf`) jusqu'à obtenir une réponse
- **exit** pour quitter

Exercice 19 (interrogations DNS avec nslookup)

Sur un terminal de votre PC, afficher le manuel en ligne de **nslookup** en tapant :

```
$ man nslookup
```

afin de s'y référer éventuellement.

Depuis un autre terminal du PC, utiliser **nslookup** pour :

1. Obtenir les serveurs SMTP en charge du domaine greenpeace.org
2. Obtenir l'adresse IPv4 de www.wwf.fr
3. quitter nslookup en tapant exit.

> ⓘ Normalement, on devrait pouvoir intérroger d'autres serveurs DNS ce qui permettrait des manipulations plus interressantes, notamment itératives. Malheureusement, un firewall à l'université bloque les requêtes/réponses.

[Corrigé]

4.2.5 WHOIS : informations sur les gestionnaires d'un domaine

En formulant une "requête whois" à une "autorité compétente", on obtient en réponse un certain nombre de renseignements sur les gestionnaires d'un nom de domaine.

L'autorité compétente pour les domaines se terminant par **.fr** et **.re** est l'AFNIC. C'est une association chargée d'attribuer et de gérer les domaines **.fr** et **.re**. Elle fournit la possibilité d'interroger en ligne sa base de données via l'URL http://www.afnic.fr/outils/whois.

Un grand nombre de domaines tels que **.arpa**, **.biz**, **.com**, **.edu**, **.org**, et autres sont gérés par l'Internic bien que **.com** et **.net** soient en fait l'exclusivité de la société VeriSign. On peut aussi interroger en ligne ces gestionnaires via l'URL http://www.internic.net/whois.html.

Enfin, on peut utiliser un site comme http://network-tools.com qui regroupe plusieurs des services que nous avons étudiés au cours de ce TP.

> ✍ Sous Linux, on peut interroger les serveurs *whois* en utilisant la commande **whois**.

Exercice 20 (interrogation whois sur Linux)

Utiliser la commande **whois** sur allegro pour obtenir des renseignements sur différents domaines tels que : **univ-aix.fr.**, **univ-mrs.fr.**, **free.fr.**, **wanadoo.fr.**, **gouv.fr.**, et autres domaines de votre choix.

[Corrigé]

5 Configuration d'un serveur et d'un client DHCP

5.1 Configuration d'un serveur DHCP

Un serveur DHCP doit avoir une interface configurée de manière statique. C'est le cas de la machine virtuelle m2 dont l'interface `eth0` est configurée statiquement avec l'adresse `10.0.2.10/24`.

> ⓘ La configuration de m1 n'est que temporaire et disparaît si on la redémarre.

Nous allons configurer m2 pour servir de serveur DHCP dans le réseau `10.0.2.0/24`. Un serveur DHCP (**dhcp3-server**) est déjà installé sur m2. Il reste à le configurer, ce qui nécessite les étapes suivantes (ne rien changer pour le moment) :

- modification du fichier `/etc/default/dhcp3-server` pour indiquer sur quelle interface le serveur répondra aux requêtes ;
- modification du fichier `/etc/dhcp3/dhcpd.conf` pour indiquer les paramètres DHCP du serveur ;
- démarrage du serveur.

Exercice 21 (configuration du serveur DHCP de m2)

Sur la machine virtuelle m2 :

1. Utiliser **vi** pour éditer le fichier `/etc/default/dhcp3-server` et indiquer `eth0` comme interface à utiliser puis quitter en sauvant ;
2. Utiliser **vi** pour éditer le fichier `/etc/dhcp3/dhcpd.conf`. Celui-ci contient déjà de nombreuses informations dont la plupart sont commentées :
   (a) modifier l'option **`domain-name`** pour contenir le domaine `iut.univ-aix.fr` ;
   (b) modifier l'option **`domain-name-servers`** pour utiliser le serveur de noms `139.124.1.2` ;
   (c) Décommenter la ligne :

   ```
   #authoritative;
   ```

   en supprimant le **#** du début ;
   (d) Mettre en commentaires la ligne :

   ```
   log-facility local7;
   ```

   en insérant un **#** du début ;
   (e) Saisir les lignes suivantes qui autorisent le serveur à allouer dynamiquement les adresses `10.0.2.100` à `10.0.2.150` dans le réseau `10.0.2.0/24` :

   ```
   subnet 10.0.2.0 netmask 255.255.255.0 {
    range 10.0.2.100 10.0.2.150;
    option routers 10.0.2.2;
    option subnet-mask 255.255.255.0;
    option broadcast-address 10.0.2.255;
   }
   ```

   (f) Quitter l'édition du fichier en le sauvant ;
3. Démarrer le serveur en tapant :

   ```
   # /etc/init.d/dhcp3-server start
   ```

[Corrigé]

5.2 Configuration d'un client DHCP

Nous allons maintenant configurer m1 comme un client DHCP (de façon temporaire).

Exercice 22 (configuration du serveur DHCP de m2)

Sur la machine virtuelle m1 :

1. Taper simplement :

   ```
   # dhclient eth0
   ```

   et patienter. Elle devrait obtenir l'adresse `10.0.2.100` qui est la première adresse allouable par le serveur.
2. Afficher la configuration IP de l'interface `eth0`, la table de routage, et le fichier `/etc/resolv.conf`, puis vérifier que tout fonctionne normalement pour m1.

[Corrigé]