# **Enoncé du TP 5 Réseaux**

## Commandes réseau de configuration et tests IP

## C. Pain-Barre

INFO - IUT Aix-en-Provence

version du 18/3/2013

## Table des matières

1	Con	figuration d'un hôte sous Unix	2					
	1.1	Les interfaces réseau	2					
	1.2	ifconfig: configuration d'une interface sous Unix	3					
		Exercice 1 (Consultation et configuration des interfaces)	5					
	1.3	Test de connectivité avec ping	7					
		Exercice 2 (tests de connectivité depuis m2)						
		Exercice 3 (tests de connectivité depuis m1)	7					
	1.4	Configuration d'une table de routage	8					
		Exercice 4 (Consultation de la table de routage de m2)						
		Exercice 5 (Modification de la table de routage de m1)	10					
		Exercice 6 (routes du PC)						
	1.5	Consultation et modification du cache ARP						
		Exercice 7 (manipulation du cache ARP)	11					
		Exercice 8 (petites questions sur arp)	12					
	1.6	traceroute : connaître le chemin suivi par les datagrammes	12					
		Exercice 9 (traceroute sur le PC)	12					
2	Con	Commandes réseau dans l'environnement Windows						
	2.1	ipconfig	14					
	2.2	netsh						
	2.3	ping	18					
	2.4	arp						
	2.5	route	19					
	2.6	tracert	22					
		Exercice 10 (Configuration de la machine virtuelle Windows XP)						
3	Synt	thèse de l'utilisation des commandes réseau	24					
	•	rcice 11 (Déduction de la topologie)	24					

## 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 :

- 1e0 pour la première ;
- le1 pour la seconde;
- etc.



Une exception concerne l'interface loopback (boucle locale) identifiée par lo suivi éventuellement d'un numéro. C'est une interface virtuelle associée à l'adresse IP 127.0.0.1 (en fait à toutes les adresses IP commençant par 127, qui sont inutilisables pour adresser un hôte sur Internet).

Virtuelle veut dire que **loopback n'est rattachée à aucune une carte réseau réelle**. Elle permet, avec son adresse IP, de tester uniquement en local des programmes utilisant TCP/IP, sans même disposer d'une liaison réseau. Elle permet donc de réaliser des tests sans pour autant provoquer de transmission sur le réseau, et d'utiliser localement des services réseaux.

Généralement, les stations sont configurées pour que cette interface puisse être désignée par le nom localhost (cas des stations Unix et Windows).

Une interface possède (généralement) une adresse physique, communément appelée adresse MAC. C'est l'adresse qui l'identifie dans les communications au niveau trame (couche liaison OSI), au sein du réseau physique auquel elle est connectée. 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 :

- son adresse IP;
- son masque de sous-réseau;



le système déduit l'adresse de son réseau en appliquant le masque à l'adresse IP, et ajoute automatiquement la route directe correspondante dans la table de routage.



- 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 la diffusion est 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 garder toutes les trames reçues sur cette interface, 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** (interface **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 ;



permet d'utiliser une autre adresse MAC (Ethernet) que celle de la carte • hw ether adresse-mac réseau. Sauf cas particuliers, il n'est pas nécessaire de préciser ce paramètre.



(i) En ligne de commandes sous Windows, la configuration d'une interface passe par la commande netsh interface ip (voir section 2.2 page 16), bien qu'ipconfig (voir section 2.1 p. 14) 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
         Link encap: Boucle locale
10
          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 10 (loopback) sont actives. Pour eth0, on peut noter la configuration mise en gras sur l'exemple :
  - adresse MAC (*HWaddr*): 00:21:9b:dd:db:f9
  - adresse IPv4 (inet adr): 139.124.187.55
  - adresse de diffusion dirigée dans le réseau (*Bcast*): 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
          . . .
10
          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 vir-
      tuelle 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. Exécuter le script /commun/pain-barre/reseaux/tp5\_mkxp4mario.bash qui prépare et démarre une machine virtuelle sous Windows XP dont on se servira plus tard. L'exécution de ce script prend du temps : passer à la suite et de temps en temps répondre aux messages de VirtualBox;
- 2. Télécharger le fichier tp5\_confip\_lab1.mar (depuis le site de l'enseignement)
- 3. Lancer le simulateur **Marionnet** via le menu *Autres*  $\longrightarrow$  *Marionnet*
- 4. Cliquer sur la fenêtre de bienvenue puis charger le fichier tp5\_confip\_lab1.mar par le menu *Projet Ouvrir* (prend un temps plus ou moins long)
- 5. 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;
  - la machine virtuelle windows XP, non configurée, représentée par B1;
  - 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 \$1 qui connecte tout ce beau monde.



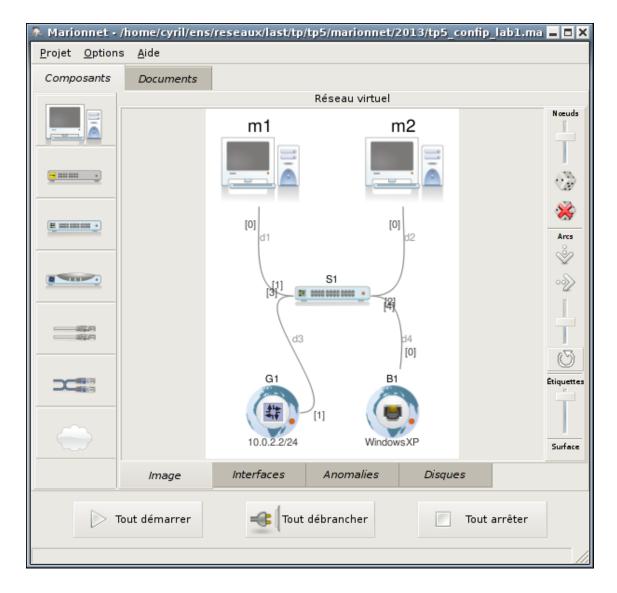


FIGURE 1 - Interface de Marionnet suite au chargement de tp5\_confip\_lab1.mar

- 6. Cliquer sur le bouton *Tout Démarrer* en bas à gauche de l'interface. Les machines virtuelles démarrent chacune dans une fenêtre terminal dédiée.
- 7. Se loger sur m2 en tant que root (mot de passe **root**)
- 8. 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
- 9. Se loger sur m1 en tant que root (mot de passe **root**)
- 10. Faire afficher la configuration de ses interfaces
- 11. Configurer son interface eth0 avec l'adresse 10.0.2.15/24
- 12. 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 18). 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.

**①** 

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 est affiché 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 si la route est directe);
- l'interface à utiliser pour cette route (lo, eth0,...), et permettant de communiquer avec la destination (route directe) ou le routeur.

La configuration de la table de routage se fait au moyen de la commande **route** sur Unix et Windows (voir section 2.5, page 19). 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
```

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	10
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 l'option **-n** pour faire afficher sa table de routage. Recommencer ensuite sans l'option **-n**.

[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 18).



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	ŀ'lags Mask	liace
b122.iut.univ-aix.fr	ether	f0:4d:a2:24:24:5a	С	eth0
imp-a.iut.univ-aix.f	r	(incomplete)		eth0
b118.iut.univ-aix.fr	ether	f0:4d:a2:23:6a:e9	С	eth0
pcdl.iut.univ-aix.fr	ether	b8:ac:6f:30:7d:65	С	eth0
<pre>iutextreme.iut.univ-</pre>	aix ether	00:23:89:57:5e:a3	С	eth0
<pre>\$ /usr/sbin/arp -n</pre>				
Address	HWtype	HWaddress	Flags Mask	Iface
139.124.187.122	ether	f0:4d:a2:24:24:5a	С	eth0
139.124.187.94		(incomplete)		eth0
139.124.187.118	ether	f0:4d:a2:23:6a:e9	С	eth0
139.124.187.248	ether	b8:ac:6f:30:7d:65	С	eth0

• • •

139.124.187.1

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.

00:23:89:57:5e:a3

#### **Exercice 7 (manipulation du cache ARP)**

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

ether

#### \$ man arp

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

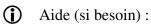
Puis, sur la machine virtuelle m1:

- 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



eth0

5. taper ping -c 1 10.0.2.10. Vous devriez ne pas obtenir de réponse. Pouvez-vous expliquer pourquoi?



- 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?
  - (i) 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 22).

**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)**

**①** 

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 infodoc.iut.univ-aix.fr
- 2. pour atteindre www.free.fr. Attention, ce peut être assez long!
  - (i) 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]

**(i)** 

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. Elle 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  $D\acute{e}marrer \longrightarrow Ex\acute{e}cuter \longrightarrow cmd$ .



### 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
```

```
Nom de l'hôte . . . . . : vb-xp-iut
Suffixe DNS principal . . . . : 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
```

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:

• 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.
            - Supprime une entrée de configuration d'une table.
delete
              - Affiche un script de configuration.
dump
help
              - Affiche une liste de commandes.
              - Réinitialisez TCP/IP et les composants associés.
reset
              - Définit l'information de configuration.
set
show
               - Affiche les informations.
Pour consulter l'aide d'une commande, entrez la commande, suivie par un espace,
```

• 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 (le tout sur une seule ligne):

```
C:>netsh int ip set address "Connexion au réseau local"
static 10.0.2.100 255.255.255.0 10.0.2.2 20
Ok.
```

• Vérifications de la configuration :

et ensuite entrez ?.

```
C:>ipconfig
```

```
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
```

• 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:
                  Envoie la requête ping sur l'hôte spécifié jusqu'à
   -t.
                  interruption.
                  Entrez Ctrl-Attn pour afficher les statistiques et continuer,
                  Ctrl-C pour arrêter.
                  Recherche les noms d'hôte à partir des adresses.
                  Nombre de requêtes d'écho à envoyer.
   -n échos
• 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]
                 Affiche les entrées ARP en cours en interrogeant les données
   -a
                  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.
                                                .... Affiche la table ARP.
   > arp -a
• Afficher le cache 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 :
                           Affiche un itinéraire
                 PRINT
                           Ajoute un itinéraire
                 ADD
                 DELETE
                           Supprime un itinéraire
                           Modifie un itinéraire existant
                 CHANGE
  destin
              Spécifie l'hôte destination.
              Si le mot clé MASK est présent, le paramètre qui le suit est
 MASK
```

```
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 déf	aut: 10.	0.2.2		

\_\_\_\_\_

Itinéraires persistants :

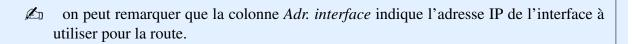
Aucun

- 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).

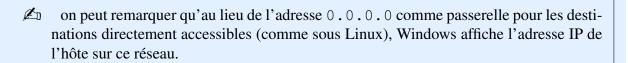
On pourra faire référence à une interface en utilisant son numéro en hexadécimal (en laissant le 0x) ou en décimal.

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 1<sup>re</sup> ligne correspond à la route par défaut qui passe par le routeur (passerelle) 10.0.2.2;



♦ la 2<sup>e</sup> ligne correspond à la route directe vers le réseau local ;



- ♦ la 3º ligne correspond à l'adresse IP de l'hôte. Elle indique que si l'hôte doit discuter avec luimême, il doit utiliser l'interface loopback;
- ♦ la 4<sup>e</sup> ligne correspond à l'adresse de diffusion dirigée dans le réseau local ;
- ♦ la 5<sup>e</sup> ligne correspond à l'interface *loopback* et regroupe toutes les adresses commençant par 127;
- ♦ la 6º 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.

20

20

20

1

1

```
• 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
                                10.0.2.1
        0.0.0.0
                    0.0.0.0
                                           10.0.2.15
                                                        2
       0.0.0.0 0.0.0.0
10.0.2.0 255.255.255.0
                               10.0.2.15
                                           10.0.2.15
                                                       20
```

Passerelle par défaut : 10.0.2.1

255.255.255.255 255.255.255

10.255.255.255 255.255.255

10.0.2.15 255.255.255.255

\_\_\_\_\_

255.0.0.0

240.0.0.0

127.0.0.1

10.0.2.15

127.0.0.1

10.0.2.15

10.0.2.15

127.0.0.1

10.0.2.15

127.0.0.1

10.0.2.15

10.0.2.15

Itinéraires persistants :

127.0.0.0

224.0.0.0

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
```

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
                          1 ms 193.50.131.18
        2 ms
                 1 ms
   6
                  2 ms
                          2 ms 193.50.131.25
        3 ms
   7
                  2 ms
                           1 ms 139.124.196.119
         3 ms
 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 192.168.100.33
        2 ms
                 4 ms
        37 ms
                          2 ms 192.168.100.2
   6
                 2 ms
   7
        15 ms
                 43 ms
                          50 ms vl10-gi8-2-marseille1-rtr-021.noc.renater.fr [193.5
   8
        12 ms
                9 ms
                          14 ms
                                tel-1-marseille2-rtr-021.noc.renater.fr [193.51.179
   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
        21 ms
  11
                          10 ms tel-1-paris2-rtr-021.noc.renater.fr [193.51.189.9]
                 13 ms
  12
       10 ms
                13 ms
                          11 ms 193.51.182.197
                          17 ms
                                72.14.238.228
  13
        23 ms
                 14 ms
  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.
```

#### Exercice 10 (Configuration de la machine virtuelle Windows XP)

Dans cet exercice, nous allons configurer la machine virtuelle Windows XP qui est représentée dans **Marionnet** par l'équipement B1 :

- 1. Se loger sur la VM Windows XP avec l'utilisateur proposé (sans mot de passe et avec les droits d'Administrateur)
- 2. Ouvrir une invite de commandes MS-DOS via le menu  $D\acute{e}marrer \longrightarrow Ex\acute{e}cuter \longrightarrow cmd$ ;
- 3. Utiliser la (ou les) commande(s) appropriée(s) pour configurer cette VM dans le réseau de m1 et m2, avec l'adresse 10.0.2.100, sans omettre le masque et le routeur par défaut (utiliser 20 comme métrique);
- 4. Utiliser les commandes appropriées pour afficher la configuration IP et la table de routage. Remarquer que la table sous Windows est assez différente de sous Linux;

- 5. Tester la connectivité de cette VM avec m2 (10.0.2.10), puis avec le routeur 10.0.2.2. Tout devrait fonctionner sinon revoir la configuration;
- 6. Afficher l'état du cache ARP de cette VM. Il devrait y avoir les adresses MAC de m2 et du routeur ;
- 7. Tester la connectivité de cette VM avec allegro (139.124.187.4). Cela devrait fonctionner;

[Corrigé]

## 3 Synthèse de l'utilisation des commandes réseau

#### Exercice 11 (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 \dots 00 07 e9 83 0f 6b \dots 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
                                                               2.0
    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
                                                               2.0
                                  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
                               130.26.148.10
                                              130.26.148.10
                                                               20
      224.0.0.0
                    240.0.0.0
 255.255.255.255 255.255.255.255
                               130.26.148.10
                                              130.26.148.10
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 ms <1 ms <1 ms 130.26.149.20
 2
                     1 ms 130.26.68.250
      1 \text{ ms}
              1 ms
 3
                    1 ms 130.26.80.200
      1 ms
             1~\mathrm{ms}
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
10
     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	10	
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
10
     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
                               255.255.240.0
130.26.80.0
              0.0.0.0
                                              U
                                                   0
                                                           0
                                                                    0 eth0
127.0.0.0
                                                                     0 10
               0.0.0.0
                               255.0.0.0
                                              U
                                                     0
                                                            0
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
   130.26.86.110 (130.26.86.110) 0.656 ms 0.986 ms 1.387 ms
```

#### Travail à faire :

130.26.64.1 (130.26.64.1) 1.847 ms

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).

2.572 ms

[Corrigé]