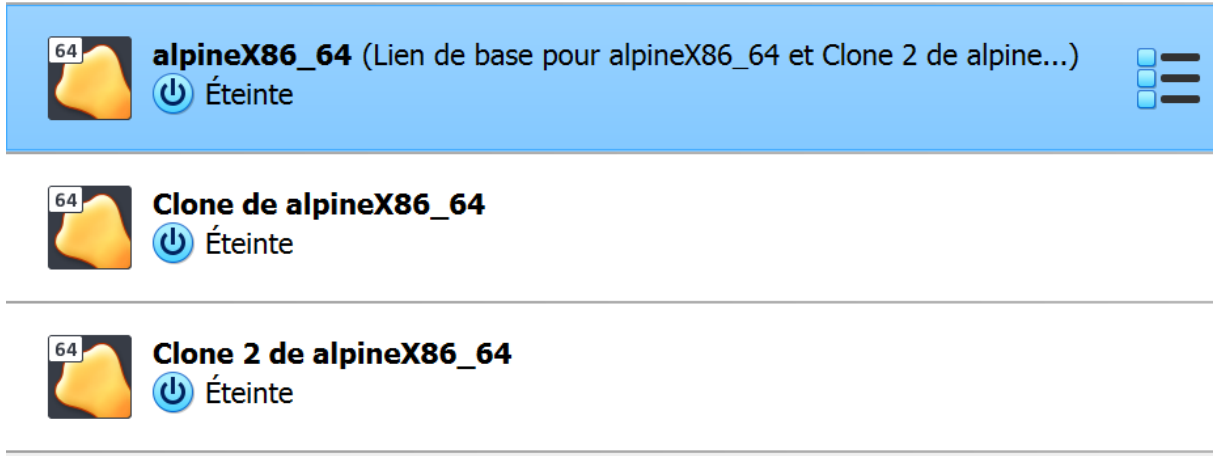


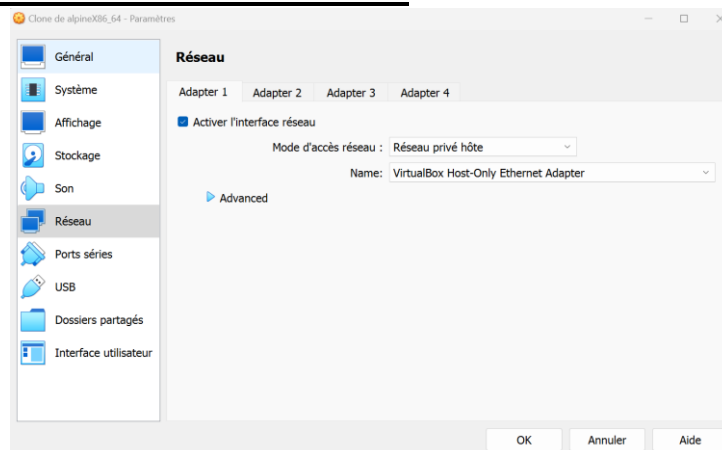
R3.06- Architecture Réseaux TP3 :

Importation et clonage des VM :



The screenshot shows the Virtual Machine Manager interface. At the top, there is a list of VMs: 'alpineX86_64' (Lien de base pour alpineX86_64 et Clone 2 de alpine...), 'Clone de alpineX86_64', and 'Clone 2 de alpineX86_64'. Each VM has a power button icon and the text 'Éteinte' (Off). The interface is in French.

- 1. Créer un réseau privé ou un réseau NAT pour les interfaces réseau afin que les VM soient dans le réseau local.**




- 2. A modifier les adresses MAC des clones**

▼ Advanced

Type d'interface : Intel PRO/1000 MT Desktop (82540EM) ▼

Mode Promiscuité : Refuser ▼

Adresse MAC : 08002785FF19 

☒ Câble branché

Préparation :

Quelle est le rôle du protocole ARP :

Le protocole ARP (Address Resolution Protocol) sert à faire une correspondance logique entre l'IP (couche réseau) et l'adresse mac (couche liaison) et il permet à une personne sur un réseau local de communiquer avec un autre individu du réseau car il lui faut l'adresse mac pour lui envoyer des trames.

Quelle est le rôle du protocole ICMP :

Le protocole ICMP (Internet Control Message Protocol) est utilisé pour envoyer des messages d'erreur et des opérations des diagnostic sur un réseau.

Plan d'adressage IP :

PC1 : 123.110.0.1/16

PC2 : 123.110.0.2/16

PC3 : 123.110.0.3/16

Choses à faire :

Vérifiez que vos 3 PC virtuels Debian sont bien raccordés au même vswitch (réseau interne ou nat).

PC1 :

Réseau

Adapter 1 Adapter 2 Adapter 3 Adapter 4

☒ Activer l'interface réseau

Mode d'accès réseau : Réseau privé hôte

Name: VirtualBox Host-Only Ethernet Adapter

PC2 :

Réseau

Adapter 1 Adapter 2 Adapter 3 Adapter 4

☒ Activer l'interface réseau

Mode d'accès réseau : Réseau privé hôte

Name: VirtualBox Host-Only Ethernet Adapter

PC3 :

Réseau

Adapter 1 Adapter 2 Adapter 3 Adapter 4

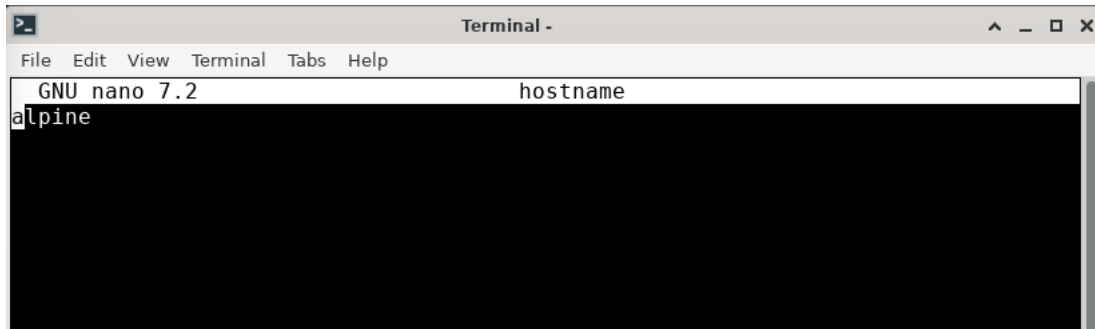
☒ Activer l'interface réseau

Mode d'accès réseau : Réseau privé hôte

Name: VirtualBox Host-Only Ethernet Adapter

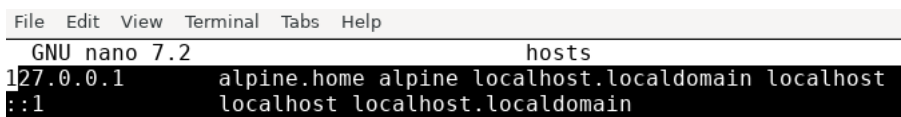
Pour éviter des problèmes de lenteur, vérifiez la correspondance du nom dans les fichiers /etc/hostname et /etc/hosts.

Hostname :



```
Terminal -
File Edit View Terminal Tabs Help
GNU nano 7.2 hostname
alpine
```

Hosts :

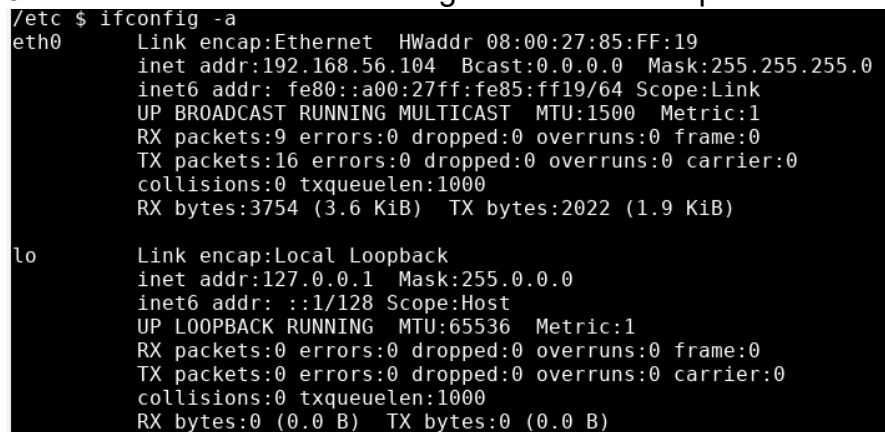


```
File Edit View Terminal Tabs Help
GNU nano 7.2 hosts
127.0.0.1 alpine.home alpine localhost.localdomain localhost
::1 localhost localhost.localdomain
```

Toutes les VM sont configuré de cette manière il n'y a donc pas de problèmes

Vérifiez que la carte réseau utilisée dans chaque PC virtuelle est bien eth0. Configurez les adresses IP statiquement (voir le plan d'adressage).

J'ai utilisé la commande ifconfig -a voici un exemple :



```
/etc $ ifconfig -a
eth0      Link encap:Ethernet  HWaddr 08:00:27:85:FF:19
          inet addr:192.168.56.104  Bcast:0.0.0.0  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe85:ff19/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:9 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3754 (3.6 KiB)  TX bytes:2022 (1.9 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Configurer les adresses fichier « interfaces »:

Voici un exemple :

```
PC1 (Lien de base pour alpineX86_64 et Clone 2 de alpineX86_64) [En fonction]
Fichier  Machine  Écran  Entrée  Périphériques  Aide
Applications  Terminal -
Terminal -
File Edit View Terminal Tabs Help
~ $ ping 123.110.0.2
PING 123.110.0.2 (123.110.0.2): 56 data bytes
64 bytes from 123.110.0.2: seq=0 ttl=42 time=2.685 ms
64 bytes from 123.110.0.2: seq=1 ttl=42 time=2.878 ms
64 bytes from 123.110.0.2: seq=2 ttl=42 time=4.061 ms
64 bytes from 123.110.0.2: seq=3 ttl=42 time=3.988 ms
64 bytes from 123.110.0.2: seq=4 ttl=42 time=3.121 ms
64 bytes from 123.110.0.2: seq=5 ttl=42 time=2.833 ms
64 bytes from 123.110.0.2: seq=6 ttl=42 time=4.489 ms
64 bytes from 123.110.0.2: seq=7 ttl=42 time=1.373 ms
64 bytes from 123.110.0.2: seq=8 ttl=42 time=2.233 ms
64 bytes from 123.110.0.2: seq=9 ttl=42 time=3.648 ms
64 bytes from 123.110.0.2: seq=10 ttl=42 time=1.819 ms
64 bytes from 123.110.0.2: seq=11 ttl=42 time=3.773 ms
64 bytes from 123.110.0.2: seq=12 ttl=42 time=3.185 ms
64 bytes from 123.110.0.2: seq=13 ttl=42 time=1.026 ms
64 bytes from 123.110.0.2: seq=14 ttl=42 time=0.492 ms
```

On peut voir que ça fonctionne car je peux ping mon deuxième PC

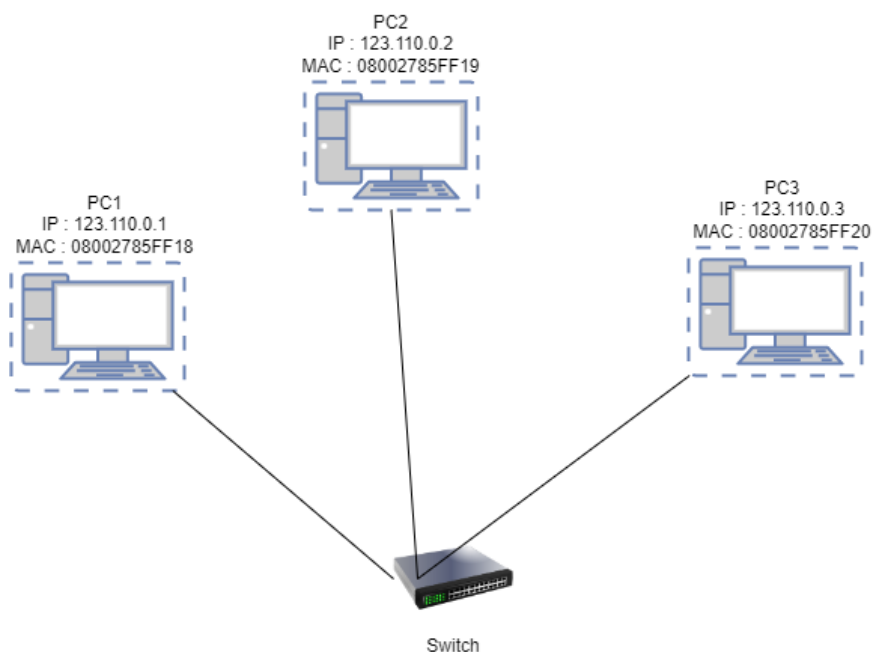
Relevez les adresses MAC des PC1, PC2 et PC3.

PC1 : 08002785FF18

PC2 : 08002785FF19

PC3 : 08002785FF20

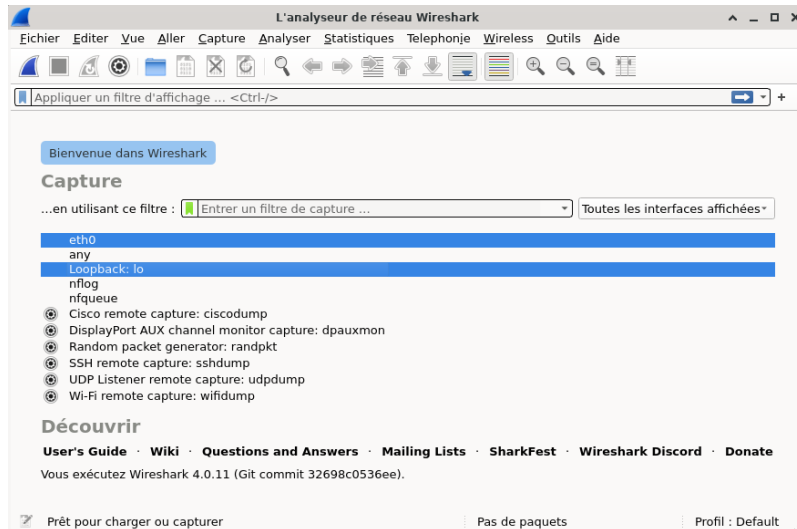
Reprenez le schéma de la maquette dans votre compte rendu et ajoutez les adresses MAC que vous avez relevées.



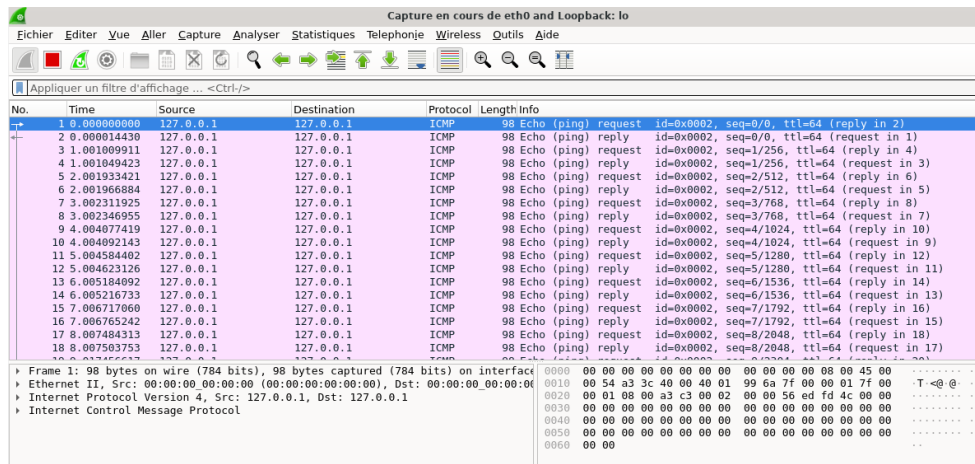
MISE EN ŒUVRE DE LA MAQUETTE

TESTS IP N°1 – ADRESSE DE LOOPBACK

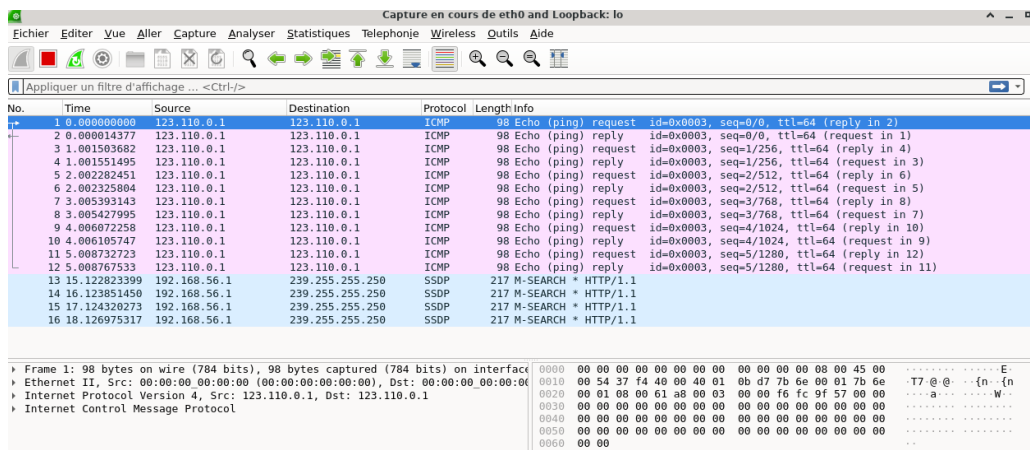
1. Lancez la capture Wireshark sur les interfaces lo et eth0 en même temps.



2. Faites un ping sur l'adresse 127.0.0.1.



3. Faites un ping sur l'adresse IPv4 du PC sur lequel vous êtes connecté actuellement.



SYNTHESE 1 :

1. Rappelez le rôle de l'adresse 127.0.0.1.

L'adresse 127.0.0.1 est l'adresse de loopback qui permet de tester la configuration de la pile réseau.

Que constatez-vous sur Wireshark :

2. En fonction de l'adresse IP, sur quelle interface le ping est visible ?

Lorsque c'est l'adresse 127.0.0.1 que nous pingons le trafic est visible sur la loopback, sur l'adresse 123.110.0.1 c'est sur eth0 que c'est visible.

3. Est-ce que le trafic généré par un PC suite à un ping vers ses adresses IP est visible sur les autres PC ?

Le trafic est visible uniquement sur le PC source.

4. Expliquez en quelques phrases simples le phénomène observé.

L'adresse loopback ne quitte jamais la machine et la deuxième est celle du PC lui-même, tout les phénomènes restent donc internes à la machine.

Test n°1

1. Videz la table arp des trois PC.
2. Depuis le PC1, lancez Wireshark et faites un ping de 4 requêtes vers l'adresse IP du PC2.
3. Lorsque deux paquets arp ont été capturés, arrêtez les captures.

Vider le cache ARP :

```
Terminal -
File Edit View Terminal Tabs Help
~ $ sudo ip link set eth0 down
[sudo] password for etudiant:
~ $ sudo ip link set eth0 up
~ $ sudo arp -n -v
~ $ arp -n -v
~ $ sudo cat /proc/net/arp
IP address      HW type    Flags      HW address    Mask        Device
~ $
```

Ping :

```
~ $ ping 123.110.0.2 -c 4
PING 123.110.0.2 (123.110.0.2): 56 data bytes
64 bytes from 123.110.0.2: seq=0 ttl=42 time=3.476 ms
64 bytes from 123.110.0.2: seq=1 ttl=42 time=5.176 ms
64 bytes from 123.110.0.2: seq=2 ttl=42 time=5.181 ms
64 bytes from 123.110.0.2: seq=3 ttl=42 time=7.729 ms

--- 123.110.0.2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 3.476/5.390/7.729 ms
```

Cache ARP du PC 1 :

```
~ $ sudo cat /proc/net/arp
[sudo] password for etudiant:
IP address      HW type    Flags      HW address    Mask        Device
123.110.0.2     0x1       0x2       08:00:27:85:ff:19  *          eth0
~ $
```

Cache ARP du PC 2 :

```
~ $ sudo cat /proc/net/arp
[sudo] password for etudiant:
IP address      HW type    Flags      HW address    Mask        Device
123.110.0.1     0x1       0x2       08:00:27:85:ff:18  *          eth0
```

Ici nous pouvons retrouver l'adresse IP et Mac de la machine source qui communique avec l'autre dans le cache ARP.

SYNTHESE 2 :

Pour le premier paquet capturé :

- Retrouvez dans le format hexadécimal, les valeurs des différents champs (entête ETHER et données ARP) du paquet.
- Dans la capture, qu'est-ce qui permet d'identifier les paquets comme étant de type ARP ?

No.	Time	Source	Destination	Protocol	Length	Info
12	83.076424771	PcsCompu_85:ff:18	Broadcast	ARP	42	Who has 123.110.0.2? Tell 123.110.0.1
13	83.078062529	PcsCompu_85:ff:19	PcsCompu_85:ff:18	ARP	60	123.110.0.2 is at 08:00:27:85:ff:19
22	88.191094864	PcsCompu_85:ff:19	PcsCompu_85:ff:18	ARP	60	Who has 123.110.0.1? Tell 123.110.0.2
23	88.191140263	PcsCompu_85:ff:18	PcsCompu_85:ff:19	ARP	42	123.110.0.1 is at 08:00:27:85:ff:18

Pour le second paquet capturé :

- Lister les différents champs de l'entête du paquet ARP et leur valeur.
- Comparez avec la capture du premier paquet.

Expliquez le rôle de chaque champ dans ARP.

Nous avons le numéro, la source qui envoie, la destination, le protocole (ARP) , la longueur et des Infos supplémentaires.

Test n°2

1. Videz les caches ARP des trois PC.
2. Sur les PC1 et PC2 :
 - a. Affichez la table arp.
 - b. Notez le résultat.
 - c. Faites un ping de 4 requêtes vers l'adresse IP du PC2.
 - d. Affichez de nouveau la table arp.
3. Depuis le PC1 :
 - a. Faites un ping de 4 requêtes vers l'adresse IP du PC3.
 - b. Affichez la table arp sur des trois PC
 - c. Notez le résultat

Les caches sont vides.

Table ARP :

PC1 :

```
Terminal -
File Edit View Terminal Tabs Help
~ $ sudo ip link set eth0 down
[sudo] password for etudiant:
~ $ sudo ip link set eth0 up
~ $ sudo arp -n -v
~ $ arp -n -v
~ $ sudo cat /proc/net/arp
IP address      HW type    Flags       HW address    Mask        Device
~ $
```

PC2 :

```
Terminal -
File Edit View Terminal Tabs Help
~ $ sudo ip link set eth0 down
[sudo] password for etudiant:
~ $ sudo ip link set eth0 up
~ $ sudo arp -n -v
~ $ arp -n -v
~ $ sudo cat /proc/net/arp
IP address      HW type    Flags       HW address    Mask        Device
~ $
```

Après le ping :

Cache ARP du PC 1 :

```
~ $ sudo cat /proc/net/arp
[sudo] password for etudiant:
IP address      HW type    Flags       HW address    Mask        Device
123.110.0.2     0x1        0x2         08:00:27:85:ff:19  *          eth0
~ $
```


Cache ARP du PC 2 :

```
~ $ sudo cat /proc/net/arp
[sudo] password for etudiant:
IP address      HW type    Flags      HW address    Mask        Device
123.110.0.1     0x1       0x2       08:00:27:85:ff:18  *          eth0
```

Après le Ping du PC3 :

PC1 :

```
~ $ sudo cat /proc/net/arp
[sudo] password for etudiant:
IP address      HW type    Flags      HW address    Mask        Device
123.110.0.2     0x1       0x2       08:00:27:85:ff:19  *          eth0
123.110.0.3     0x1       0x2       08:00:27:85:ff:20  *          eth0
```

PC2 :

```
~ $ sudo cat /proc/net/arp
[sudo] password for etudiant:
IP address      HW type    Flags      HW address    Mask        Device
123.110.0.1     0x1       0x2       08:00:27:85:ff:18  *          eth0
```

PC3 :

```
~ $ sudo cat /proc/net/arp
[sudo] password for etudiant:
IP address      HW type    Flags      HW address    Mask        Device
123.110.0.1     0x1       0x2       08:00:27:85:ff:18  *          eth0
```

On peut voir que le cache ARP contient deux entrées dans son cache pour le PC1 et 1 pour le PC2 et le PC3.

Test n°3

1. Videz la table arp des trois PC.
2. Depuis le PC2 :
 - a. Ajoutez manuellement l'adresse IP et l'adresse MAC des PC1 et PC3 dans la table arp du PC2 à l'aide de la commande donnée précédemment.
 - b. Affichez la table arp du PC2.
 - c. Notez le résultat.
3. Lancez Wireshark sur les trois PC.
4. Depuis le PC2, faites un ping de 4 requêtes vers les PC1 et PC3.
 - a. Notez les types de paquets et le nombre de trames échangées entre les PC.

Les caches sont vides.

```
~ $ sudo arp -s 123.110.0.1 08:00:27:85:ff:18
~ $ sudo cat /proc/net/arp
IP address      HW type    Flags       HW address    Mask        Device
123.110.0.1      0x1        0x6         08:00:27:85:ff:18  *          eth0
~ $ sudo arp -s 123.110.0.3 08:00:27:85:ff:20
~ $ sudo cat /proc/net/arp
IP address      HW type    Flags       HW address    Mask        Device
123.110.0.1      0x1        0x6         08:00:27:85:ff:18  *          eth0
123.110.0.3      0x1        0x6         08:00:27:85:ff:20  *          eth0
```

Ping depuis le PC2 vers le PC1 puis le PC3 :

```
Terminal -
File Edit View Terminal Tabs Help
~ $ ping 123.110.0.1 -c 4
PING 123.110.0.1 (123.110.0.1): 56 data bytes
64 bytes from 123.110.0.1: seq=0 ttl=42 time=4.166 ms
64 bytes from 123.110.0.1: seq=1 ttl=42 time=6.054 ms
64 bytes from 123.110.0.1: seq=2 ttl=42 time=5.683 ms
64 bytes from 123.110.0.1: seq=3 ttl=42 time=5.188 ms

--- 123.110.0.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 4.166/5.272/6.054 ms
~ $ ping 123.110.0.3 -c 4
PING 123.110.0.3 (123.110.0.3): 56 data bytes
64 bytes from 123.110.0.3: seq=0 ttl=42 time=4.411 ms
64 bytes from 123.110.0.3: seq=1 ttl=42 time=4.257 ms
64 bytes from 123.110.0.3: seq=2 ttl=42 time=1.690 ms
64 bytes from 123.110.0.3: seq=3 ttl=42 time=4.147 ms

--- 123.110.0.3 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1.690/3.626/4.411 ms
```

Il y a 8 trames échangé au total car il y a l'envoi du ping et la réponse.

Test n°4

1. Videz les caches arp des trois PC.
2. Sur le PC1:
 - a. Lancez wireshark
 - b. Ajoutez manuellement l'adresse IP du PC2 mais n'indiquez pas sa vraie adresse MAC mais une fausse (par exemple : 08:00:07:01:02:03).
 - c. Affichez la table arp.
 - d. Notez le résultat.
 - e. Faites un ping de 4 requêtes vers le PC2.
 - f. Observez les trames échangées dans wireshark.
3. Sur le PC2 :
 - a. Affichez la table arp.
 - b. Notez le résultat.

Les caches sont vides.

```
~ $ sudo arp -s 123.110.0.2 08:00:07:01:02:03
[sudo] password for etudiant:
~ $ sudo cat /proc/net/arp
IP address      HW type        Flags          HW address      Mask           Device
123.110.0.2      0x1            0x6            08:00:07:01:02:03  *              eth0

~ $ ping 123.110.0.2 -c 4
PING 123.110.0.2 (123.110.0.2): 56 data bytes

--- 123.110.0.2 ping statistics ---
4 packets transmitted, 0 packets received, 100% packet loss
```

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	fe80::a00:27ff:fe85::	ff02::12	ICMPv6	70	Router Solicitation from 08:00:27:85:ff:20
2	4.136110353	fe80::a00:27ff:fe85::	ff02::2	ICMPv6	70	Router Solicitation from 08:00:27:85:ff:19
3	12.875769536	fe80::a00:27ff:fe85::	ff02::2	ICMPv6	70	Router Solicitation from 08:00:27:85:ff:18
4	72.493149361	192.168.56.1	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
5	72.493150220	192.168.56.1	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
6	73.494228038	192.168.56.1	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
7	73.494229187	192.168.56.1	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
8	74.495648932	192.168.56.1	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
9	74.495649979	192.168.56.1	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
10	75.496093262	192.168.56.1	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
11	75.496094317	192.168.56.1	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
12	94.438738208	192.168.56.1	224.0.0.251	MDNS	85	Standard query 0x0000 PTR _microsoft_mcc._tcp.local, "QU" question
13	94.439546597	fe80::7b40:5e6f:b32::	ff02::fb	MDNS	105	Standard query 0x0000 PTR _microsoft_mcc._tcp.local, "QU" question
14	95.437433082	192.168.56.1	224.0.0.251	MDNS	85	Standard query 0x0000 PTR _microsoft_mcc._tcp.local, "QM" question
15	95.439014087	fe80::7b40:5e6f:b32::	ff02::fb	MDNS	105	Standard query 0x0000 PTR _microsoft_mcc._tcp.local, "QM" question
16	121.897412469	fe80::a00:27ff:fe85::	ff02::2	ICMPv6	70	Router Solicitation from 08:00:27:85:ff:19
17	129.121605007	123.110.0.1	123.110.0.2	ICMP	98	Echo (ping) request id=0x0007, seq=0/0, ttl=64 (no response found!)
18	130.124307439	123.110.0.1	123.110.0.2	ICMP	98	Echo (ping) request id=0x0007, seq=1/256, ttl=64 (no response found!)

Il n'y a aucun retour d'écho car la mac est fausse.

Test n°5

1. Videz les caches des trois PC.
2. Sur le PC2 :
 - a. Ajoutez manuellement l'adresse IP du PC1 mais indiquez l'adresse MAC du PC3
 - b. Affichez le cache ARP.
 - c. Notez le résultat.
 - d. Faites un ping vers l'adresse IP du PC1.
 - e. Notez la réponse au ping.

Les caches sont vides.

```
~ $ sudo arp -s 123.110.0.1 08:00:27:85:ff:20
~ $ sudo cat /proc/net/arp
IP address      HW type        Flags          HW address      Mask           Device
123.110.0.1      0x1            0x6            08:00:27:85:ff:20  *              eth0

~ $ sudo cat /proc/net/arp
IP address      HW type        Flags          HW address      Mask           Device
123.110.0.1      0x1            0x6            08:00:27:85:ff:20  *              eth0
~ $ ping 123.110.0.1 -c 4
PING 123.110.0.1 (123.110.0.1): 56 data bytes

--- 123.110.0.1 ping statistics ---
4 packets transmitted, 0 packets received, 100% packet loss
```

SYNTHESE 3 :

Pour le test n°2 :

Quel est l'enchaînement des actions et des échanges protocolaires qui aboutit au remplissage de la table arp des PC et à un résultat positif à un ping ?

Le protocole ARP remplit la table ARP avec les adresses MAC correspondantes aux adresses IP ciblées, permettant au ping de réussir grâce à la résolution de l'adresse physique nécessaire à la transmission des paquets ICMP.

Pour le test n°3 :

Quel changement notez-vous dans les trames capturées en comparant avec le test n°2 ? En déduire, l'usage du protocole ARP.

Les trames ARP sont moins fréquentes dans les captures postérieures au premier ping, ce qui indique que le protocole ARP évite les demandes répétitives en cachant les associations IP-MAC récemment résolues.

Pour le test n°4 :

Le PC2 est-il joignable depuis le PC1 ?

Quels changements notez-vous dans les trames capturées ?

Quels sont les PCs qui ont reçu la requête ICMP du PC1 ?

Le PC2 n'est pas joignable depuis le PC1 en raison d'une entrée ARP incorrecte, ce qui entraîne des requêtes ICMP sans réponse visible dans les trames capturées par Wireshark.

Pour le test n°5 :

Quel PC reçoit les requêtes ICMP du PC2 et lequel y répond ?

Expliquez pourquoi ?

Seul le PC ciblé par les requêtes ICMP les reçoit et y répond, ce qui démontre la nécessité d'une résolution ARP correcte pour une communication inter-réseau efficace.