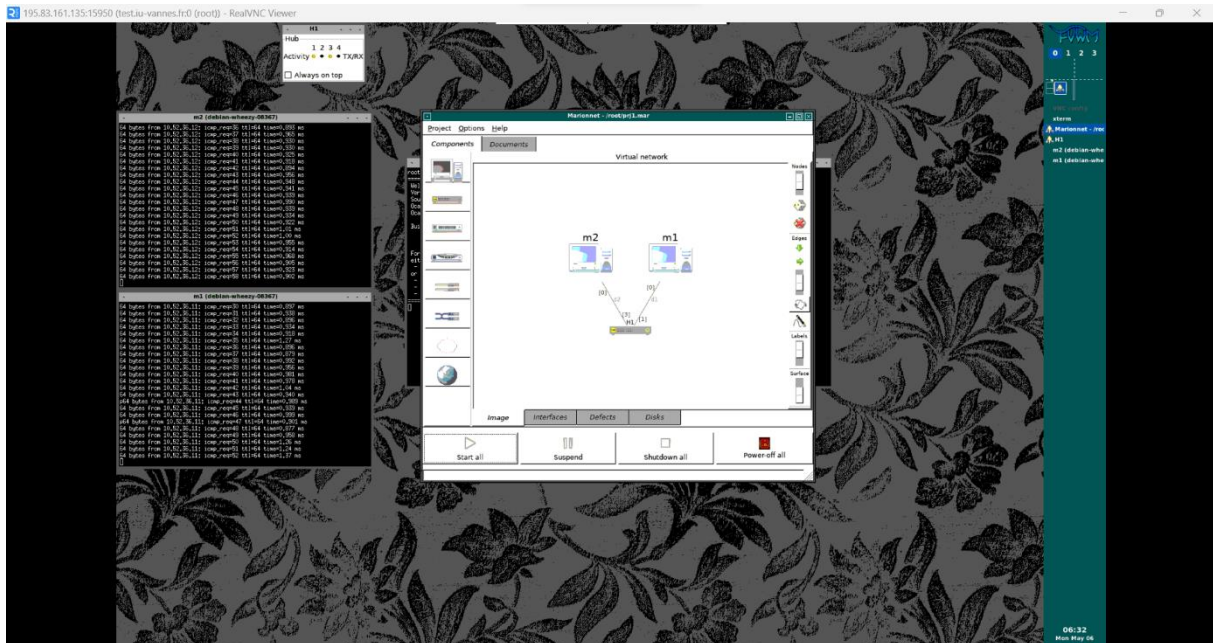


R2.05 – Services Réseaux

TP3

Section 1 :



Pour vérifier que les deux machines communiquent ensemble j'ai utilisé la commande « ping [adresse ip de l'autre machine] ».

Sur m2 j'ai tapé : ping 10.52.36.12

Sur m1 j'ai tapé : ping 10.52.36.11

```
m1 (debian-wheezy-08367)
64 bytes from 10.52.36.11: icmp_req=254 ttl=64 time=0.999 ns
64 bytes from 10.52.36.11: icmp_req=255 ttl=64 time=0.933 ns
64 bytes from 10.52.36.11: icmp_req=256 ttl=64 time=0.943 ns
64 bytes from 10.52.36.11: icmp_req=257 ttl=64 time=1.01 ns
64 bytes from 10.52.36.11: icmp_req=258 ttl=64 time=1.52 ns
64 bytes from 10.52.36.11: icmp_req=259 ttl=64 time=1.43 ns
64 bytes from 10.52.36.11: icmp_req=260 ttl=64 time=1.53 ns
64 bytes from 10.52.36.11: icmp_req=261 ttl=64 time=1.40 ns
64 bytes from 10.52.36.11: icmp_req=262 ttl=64 time=0.953 ns
64 bytes from 10.52.36.11: icmp_req=263 ttl=64 time=0.992 ns
64 bytes from 10.52.36.11: icmp_req=264 ttl=64 time=0.940 ns
64 bytes from 10.52.36.11: icmp_req=265 ttl=64 time=0.963 ns
64 bytes from 10.52.36.11: icmp_req=266 ttl=64 time=0.969 ns
64 bytes from 10.52.36.11: icmp_req=267 ttl=64 time=0.950 ns
64 bytes from 10.52.36.11: icmp_req=268 ttl=64 time=0.989 ns
64 bytes from 10.52.36.11: icmp_req=269 ttl=64 time=1.01 ns
^C
--- 10.52.36.11 ping statistics ---
269 packets transmitted, 269 received, 0% packet loss, time 269923ms
rtt min/avg/max/ndev = 0.804/1.075/1.695/0.220 ms
[0 root@m1 ~]$ arp
Address          Hwtype Hwaddress      Flags Mask      Iface
10.52.36.11      ether   02:04:06:23:3a:15 C                eth0
[0 root@m1 ~]$
```

```
m2 (debian-wheezy-08367)
64 bytes from 10.52.36.12: icmp_req=263 ttl=64 time=0.865 ns
64 bytes from 10.52.36.12: icmp_req=264 ttl=64 time=0.960 ns
64 bytes from 10.52.36.12: icmp_req=265 ttl=64 time=0.772 ns
64 bytes from 10.52.36.12: icmp_req=266 ttl=64 time=0.956 ns
64 bytes from 10.52.36.12: icmp_req=267 ttl=64 time=0.739 ns
64 bytes from 10.52.36.12: icmp_req=268 ttl=64 time=0.684 ns
64 bytes from 10.52.36.12: icmp_req=269 ttl=64 time=0.724 ns
64 bytes from 10.52.36.12: icmp_req=270 ttl=64 time=0.745 ns
64 bytes from 10.52.36.12: icmp_req=271 ttl=64 time=0.750 ns
64 bytes from 10.52.36.12: icmp_req=272 ttl=64 time=0.841 ns
64 bytes from 10.52.36.12: icmp_req=273 ttl=64 time=0.720 ns
64 bytes from 10.52.36.12: icmp_req=274 ttl=64 time=0.918 ns
64 bytes from 10.52.36.12: icmp_req=275 ttl=64 time=0.898 ns
64 bytes from 10.52.36.12: icmp_req=276 ttl=64 time=0.994 ns
64 bytes from 10.52.36.12: icmp_req=277 ttl=64 time=1.33 ns
64 bytes from 10.52.36.12: icmp_req=278 ttl=64 time=1.43 ns
^C
--- 10.52.36.12 ping statistics ---
278 packets transmitted, 278 received, 0% packet loss, time 278647ms
rtt min/avg/max/ndev = 0.679/1.005/2.241/0.218 ms
[0 root@m2 ~]$ arp
Address          Hwtype Hwaddress      Flags Mask      Iface
10.52.36.12      ether   02:04:06:d2:61:2c C                eth0
[0 root@m2 ~]$
```

Section 2 :

Le PC Espion a connaissance des paquets qui passe et leurs contenu, ici un ICMP Echo Request soit un ping.

```

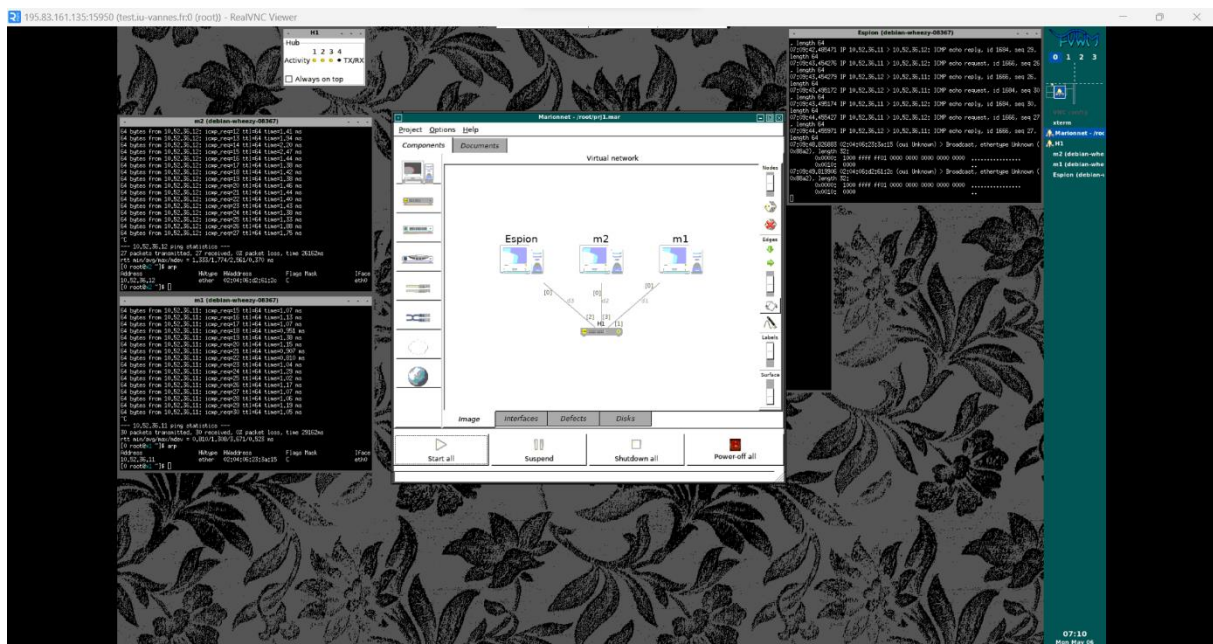
Espion (debian-wheezy-08367)
, length 64
06:40:54.969489 IP 10.52.36.11 > 10.52.36.12: ICMP echo reply, id 1633, seq 65,
length 64
06:40:55.964679 IP 10.52.36.11 > 10.52.36.12: ICMP echo request, id 1642, seq 72
, length 64
06:40:55.964682 IP 10.52.36.12 > 10.52.36.11: ICMP echo reply, id 1642, seq 72,
length 64
06:40:55.970497 IP 10.52.36.12 > 10.52.36.11: ICMP echo request, id 1633, seq 66
, length 64
06:40:55.970500 IP 10.52.36.11 > 10.52.36.12: ICMP echo reply, id 1633, seq 66,
length 64
06:40:56.375221 IP 10.52.36.11 > 10.52.36.12: ICMP echo request, id 1642, seq 73
, length 64
06:40:56.375232 IP 10.52.36.12 > 10.52.36.11: ICMP echo reply, id 1642, seq 73,
length 64
06:40:56.973098 IP 10.52.36.12 > 10.52.36.11: ICMP echo request, id 1633, seq 67
, length 64
06:40:56.973530 IP 10.52.36.11 > 10.52.36.12: ICMP echo reply, id 1633, seq 67,
length 64
06:40:57.378531 IP 10.52.36.11 > 10.52.36.12: ICMP echo request, id 1642, seq 74
, length 64
06:40:57.378534 IP 10.52.36.12 > 10.52.36.11: ICMP echo reply, id 1642, seq 74,
length 64

```

Concernant les diodes, les 3 diodes des différents port clignote en même temps ce qu'il veut dire qu'il reçoit des informations de la part du hub en même temps.

Avant tout échange entre deux machine, il va y avoir un premier échange entre les machines pour qu'ils s'enregistrent mutuellement dans leurs table ARP respective.

Il y a uniquement l'ARP de la deuxième machine qui est vu par la machine Espion, pour cause la machine m1 possède déjà dans sa table ARP l'adresse de m2 mais m2 ne possède pas l'ARP de la machine m1 vu qu'on la supprimer.



Section 3 :

Selon Wikipédia MTU veut dire : Lors d'une transmission de données informatiques, la maximum transmission unit est la taille maximale d'un paquet pouvant être transmis en une seule fois sur une interface. Donc en résumé MTU c'est la quantité d'information que peut être envoyé en un seul paquet.

La commande utilisé pour faire des paquets de taille souhaité on utilise la commande « ping -s [taille paquet] [adresse ip de la machine]

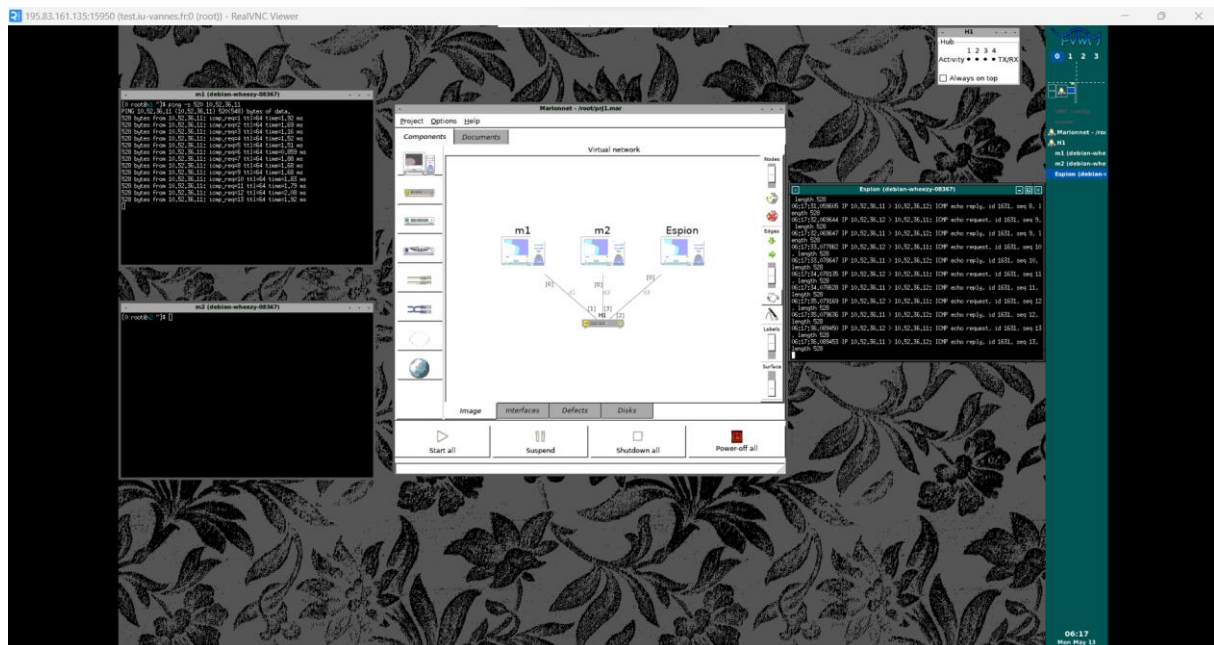
```
[root@red ~]# ping -s 65525 10.52.36.11
WARNING: packet size 65525 is too large, Maximum is 65507
PING 10.52.36.11 (10.52.36.11) 65525(65553) bytes of data.
ping: local error: Message too long, mtu=16436
ping: local error: Message too long, mtu=16436
ping: local error: Message too long, mtu=16436
ping: local error: Message too long, mtu=16436
Xping: local error: Message too long, mtu=16436
ping: local error: Message too long, mtu=16436
ping: local error: Message too long, mtu=16436
ping: local error: Message too long, mtu=16436
^C
--- 10.52.36.11 ping statistics ---
8 packets transmitted, 0 received, +8 errors, 100% packet loss, time 7009ms
```

Taille maximum d'un paquet est de 65507, le même message apparait dans les deux cas, lorsqu'on tente de m1 vers m2 et de m2 vers m1. Il y a une fragmentation des paquets pour pouvoir faciliter l'envoi de message, il va être plus simple d'envoyer des petit paquet pour avoir un envoie continue au client que plutôt essayer de tout envoyer car l'envoi d'un paquet complet va prendre plus de temps et pendant ce temps la personne qui reçoit le paquet n'a aucune information de ce qui se passe.

En tenant compte de l'en-tête IP (généralement de 20 octets) et des paquets ICMP de 500 octets, nous pouvons calculer le MTU.

- MTU minimum = Taille du paquet ICMP + Taille de l'en-tête IP
- Pour un paquet ICMP de 500 octets et un en-tête IP de 20 octets :
- MTU minimum = 500 octets (paquet ICMP) + 20 octets (en-tête IP)
- MTU minimum = 520 octets

Ainsi, le MTU minimum pour qu'il n'y ait pas de fragmentation des paquets ICMP de 500 octets serait de 520 octets. Tout MTU supérieur à cette valeur entrainera une fragmentation des paquets ICMP.



Section 4 :

Adresse Mac de la carte réseau ETH0 : 02:04:06:9a:cb:b3

Adresse Mac de la carte réseau ETH1 : 02:04:06:32:0d:bc

Après avoir configuré l'adresse IP des deux cartes réseaux du routeur, en faisant la commande « ifconfig » on a toute les informations concernant les cartes réseaux dont l'adresse MAC.

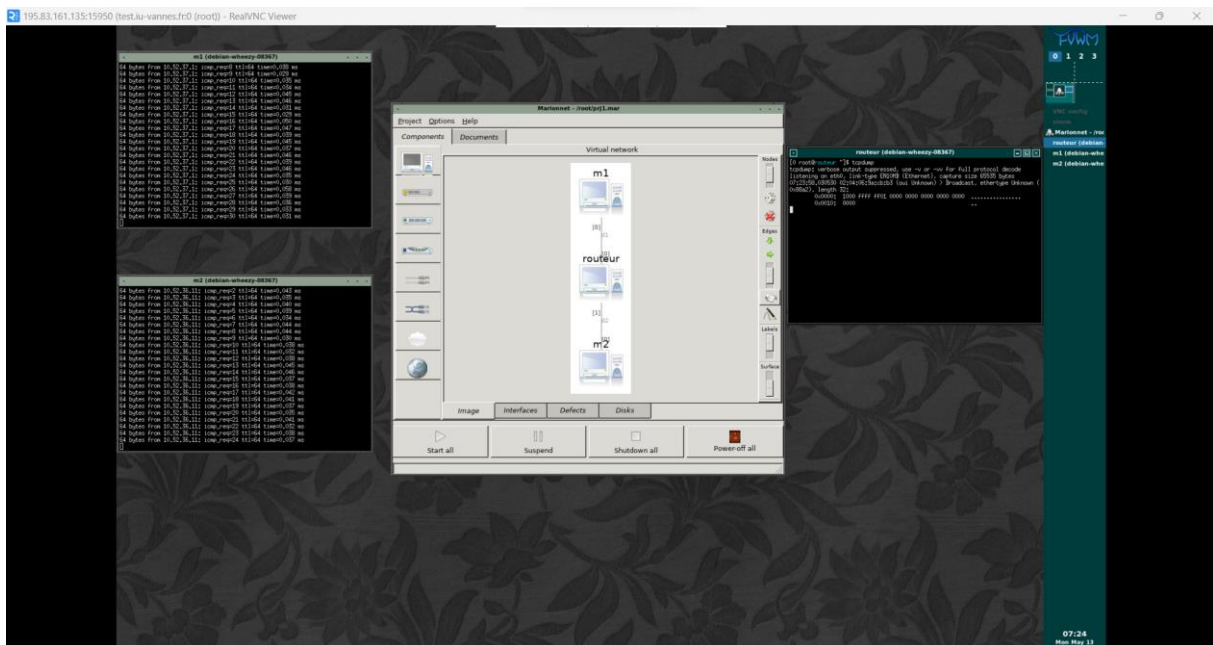
Indiquez les adresses MAC trouvées et comment vous les avez trouvé dans une section intitulée "Section 4".

Le host n'est pas atteignable via la commande ping. Cela est valable de M1 -> Routeur mais également de M2 -> Routeur.

Maintenant tester le ping de M1 à M2, que manque t-il encore ?

Après avoir tapé la commande : « `sysctl -w net.ipv4.ip_forward=1` ». Les deux machines n'arrivent pas à communiquer car il manque une table de routage. La machine M1 ne sait pas que la machine M2 est accessible via le routeur et pareil pour M2 il ne sait pas que M1 est accessible via le routeur également.

```
10.52.37.1 eth1
```



Section 5 :

```
tracert 10.52.36.11 depuis M1
```

```
tracert 10.52.36.12 depuis M2
```