

# TP 2 reseaux

## Câblage et configuration de base

Après avoir branché toutes les machines aux switches, on attribue à chaque machine une adresse IP en utilisant les commande suivantes

```
$ ifconfig eth0 192.168.1.1 netmask 255.255.255.0
```

```
$ ifconfig eth0 192.168.1.2 netmask 255.255.255.0
```

```
$ ifconfig eth0 192.168.1.3 netmask 255.255.255.0
```

Une fois les adresses IP configurées, on utilise la commande *ping* pour s'assurer que les liaisons on bien été établies

- ping réalisés à partir de la machine m1

```
$ ping 192.168.1.2
```

```
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.  
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=27.4 ms  
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.777 ms  
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=0.791 ms  
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=1.16 ms  
64 bytes from 192.168.1.2: icmp_seq=5 ttl=64 time=0.928 ms  
64 bytes from 192.168.1.2: icmp_seq=6 ttl=64 time=0.514 ms  
  
--- 192.168.1.2 ping statistics ---  
6 packets transmitted, 6 received, 0% packet loss, time 5059ms  
rtt min/avg/max/mdev = 0.514/5.271/27.456/9.923 ms
```

```
$ ping 192.168.1.3
```

```
PING 192.168.1.3 (192.168.1.3) 56(84) bytes of data.  
from 192.168.1.3: icmp_seq=1 ttl=64 time=23.6 ms  
from 192.168.1.3: icmp_seq=2 ttl=64 time=0.832 ms  
64 bytes from 192.168.1.3: icmp_seq=3 ttl=64 time=3.17 ms  
64 bytes from 192.16 8.1.3: icmp_seq=4 ttl=64 time=1.59 ms  
64 bytes from 192.168.1.3: icmp_seq=5 ttl=64 time=1.55 ms  
64 bytes from 192.168.1.3: icmp_seq=6 ttl=64 time=0.791 ms  
  
--- 192.168.1.3 ping statistics ---  
6 packets transmitted, 6 received, 0% packet loss, times 6059ms  
rtt min/avg/max/mdev = 0.791/4.680/23.621/7.768 ms
```

- ping réalisés à partir la machine m2

```
$ ping -192.168.1.1
```

```
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.  
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=12.7 ms  
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.742 ms  
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=2.72 ms  
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=0.934 ms  
64 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=0.775 ms  
64 bytes from 192.168.1.1: icmp_seq=6 ttl=64 time=0.408 ms  
  
--- 192.168.1.1 ping statistics ---  
6 packets transmitted, 6 received, 0% packet loss, time 5056ms  
rtt min/avg/max/mdev = 0.408/3.057/12.764/4.405
```

```
$ ping 192.168.1.3
```

```
PING 192.168.1.3 (192.168.1.3) 56(84) bytes of data.  
64 bytes from 192.168.1.3: icmp_seq=1 ttl=64 time=21.0 ms  
64 bytes from 192.168.1.3: icmp_seq=2 ttl=64 time=0.785 ms  
64 bytes from 192.168.1.3: icmp_seq=3 ttl=64 time=0.541 ms  
64 bytes from 192.168.1.3: icmp_seq=4 ttl=64 time=1.18 ms  
64 bytes from 192.168.1.3: icmp_seq=5 ttl=64 time=0.880 ms  
64 bytes from 192.168.1.3: icmp_seq=6 ttl=64 time=1.71 ms  
  
--- 192.168.1.3 ping statistics ---  
6 packets transmitted, 6 received, 0% packet loss, time 5044ms  
rtt min/avg/max/mdev = 0.541/4.354/21.030/7.466 ms
```

- ping réalisé à partir de la machine m3

```
$ ping 192.168.1.1
```

```
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.  
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=21.3 ms  
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.730 ms  
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.774 ms  
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=1.09 ms  
64 bytes from 192.168.1.1: icmp_seq=5 ttl=64 time=0.680 ms  
64 bytes from 192.168.1.1: icmp_seq=6 ttl=64 time=1.05 ms  
  
--- 192.168.1.1 ping statistics ---  
6 packets transmitted, 6 received, 0% packet loss, time 5043ms  
rtt min/avg/max/mdev = 0.521/0.807/1.091/0.204 ms
```

```
$ ping -c 6 192.168.1.2
```

```
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.  
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=21.2 ms  
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.832 ms  
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=0.530 ms  
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=0.826 ms  
64 bytes from 192.168.1.2: icmp_seq=5 ttl=64 time=1.24 ms  
64 bytes from 192.168.1.2: icmp_seq=6 ttl=64 time=0.582 ms  
  
--- 192.168.1.2 ping statistics ---  
6 packets transmitted, 6 received, 0% packet loss, time 5039ms  
rtt min/avg/max/mdev = 0.530/0.803/1.242/0.230 ms
```

On remarque que la première requête ping prends plus de temps que les suivantes. La raison de ce phénomène est que, juste après la configuration du réseau, la table ARP est vide donc l'émetteur ne sait pas à quelle adresse matérielle correspond l'adresse IP qu'il cherche à joindre. Ainsi, pour connaître l'adresse du destinataire, une requête ARP demandant est "broadcastée" vers toutes les machines du réseau. Une fois que la machine concernée a répondu, la connexion peut-être établie.

## Debrancher un cable

Lorsque l'on débranche un câble, par exemple, entre la machine 1 et la machine 3, les deux machines ne peuvent plus communiquer.

Dans ce cas de figure, une tentative de ping renvoie au bout de quelques instants le message suivant : *Destination Host Unreachable*.

En rebranchant le câble, les deux machines peuvent à nouveau communiquer.

## Configuration des noms symboliques

Le fichier *hosts* initial est le suivant:

```
127.0.0.1      localhost  
  
The following lines are desirable for IPv6 capable hosts  
  
:::1          ip6-localhost ip6-loopback  
fe00::0       ip6-localnet  
ff00::0       ip6-mcastprefix  
ff02::1       ip6-allnodes  
ff02::2       ip6-allrouters  
ff02::3       ip6-allhosts  
127.0.0.1     a  
127.0.0.1     m1
```

après la modification, on obtient :

```
192.168.1.2 m2 machine2
192.168.1.3 m3 machine3
```

The following lines are desirable for IPv6 capable hosts

```
::1      ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
127.0.0.1 a
127.0.0.1 m1
```

On applique la même démarche pour chaque machine

Dans le fichier *hosts* de la machine 2, on ajoute:

```
192.168.1.1 m1 machine1
192.168.1.3 m3 machine3
```

Dans le fichier *hosts* de la machine 3, on ajoute:

```
192.168.1.1 m1 machine1
192.168.1.2 m2 machine2
```

Test des modifications effectuées :

- machine 1 → machine 2

```
ping -c 2 m2
PING m2 (192.168.1.2) 56(84) bytes of data.
64 bytes from m2 (192.168.1.2): icmp_seq=1 ttl=64 time=0.524 ms
64 bytes from m2 (192.168.1.2): icmp_seq=2 ttl=64 time=1.45 ms

--- m2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.524/0.987/1.451/0.464 ms
```

- machine 1 → machine 3

```
ping -c 2 m3
PING m3 (192.168.1.3) 56(84) bytes of data.
64 bytes from m3 (192.168.1.3): icmp_seq=1 ttl=64 time=1.31 ms
64 bytes from m3 (192.168.1.3): icmp_seq=2 ttl=64 time=1.06 ms

--- m3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 1.068/1.190/1.313/0.127 ms
```

- machine 2 → machine 1

```
ping -c 2 m1
PING m1 (192.168.1.1) 56(84) bytes of data.
64 bytes from m1 (192.168.1.1): icmp_seq=1 ttl=64 time=21.2 ms
64 bytes from m1 (192.168.1.1): icmp_seq=2 ttl=64 time=0.603 ms

--- m1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1007ms
rtt min/avg/max/mdev = 0.603/10.926/21.249/10.323 ms
```

- machine 2 → machine 3

```
ping -c 2 m3
PING m3 (192.168.1.3) 56(84) bytes of data.
64 bytes from m3 (192.168.1.3): icmp_seq=1 ttl=64 time=21.2 ms
64 bytes from m3 (192.168.1.3): icmp_seq=2 ttl=64 time=1.55 ms

--- m3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1016ms
rtt min/avg/max/mdev = 1.550/11.393/21.237/9.844 ms
```

- machine 3 → machine 1

```
ping -c 2 m1
PING m1 (192.168.1.1) 56(84) bytes of data.
64 bytes from m1 (192.168.1.1): icmp_seq=1 ttl=64 time=22.1 ms
64 bytes from m1 (192.168.1.1): icmp_seq=2 ttl=64 time=0.700 ms

--- m1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1017ms
rtt min/avg/max/mdev = 0.700/11.415/22.131/10.716 ms
```

root

- machine 3 → machine 2

```
ping -c 2 m2
PING m2 (192.168.1.2) 56(84) bytes of data.
64 bytes from m2 (192.168.1.2): icmp_seq=1 ttl=64 time=1.65 ms
64 bytes from m2 (192.168.1.2): icmp_seq=2 ttl=64 time=1.14 ms

--- m2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 1.146/1.402/1.658/0.256 ms
```

On remarque que la modification du fichier *hosts* nous permet bien d'associer un identifiant à une adresse. L'option -c nbRequete de la ping permet de définir le nombre de pings que l'on va effectuer.

## ARP (Address Resolution Protocol)

après avoir effectué des ping sur m1 et m2, on obtient le résultat suivant:

```
$ arp
```

Address	Hwtype	Hwaddress	Flags	Mask	Iface
m3	ether	02:04:06:E0:EE:F5	C		eth0
m2	ether	02:04:06:FF:73:C3	C		eth0

```
$ arp -n
```

Address	Hwtype	Hwaddress	Flags	Mask	Iface
192.168.1.3	ether	02:04:06:E0:EE:F5	C		eth0
192.168.1.2	ether	02:04:06:FF:73:C3	C		eth0

d'après les commandes ci-dessus, les adresses MAC de m2 et m3 sont respectivement 02:04:06:E0:EE:F5 et 02:04:06:FF:73:C3.

la commande *ifconfig* sur les machines m2 et m3 donne:

```
m2:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:04:06:ff:73:c3
          inet addr:192.168.1.2  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::4:6ff:feff:73c3/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:57 errors:0 dropped:0 overruns:0 frame:0
          TX packets:32 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1760 (1.7 KiB)  TX bytes:1594 (1.5 KiB)
          Interrupt:5
```

```
m3:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:04:06:e0:ee:f5
          inet addr:192.168.1.3  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::4:6ff:fee0:ee:f5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:57 errors:0 dropped:0 overruns:0 frame:0
          TX packets:33 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1760 (1.7 KiB)  TX bytes:1636 (1.5 KiB)
          Interrupt:5
```

On remarque qu'il y a bien une correspondance entre les adresses MAC données par la table ARP sur m1 et celles données par *ifconfig* sur m2 et m3.

## Capture de trames ARP

Le type à l'intérieur des trames ethernet de la requête et de la réponse ARP est "ARP (0x0806)".

Le type de la requête et de la réponse ICMP est "IP (0x0800)".

Quand on sélectionne la trame *arp* dans wireshark, on voit que on a les 3 couches d'encapsulation suivantes :

- La 1ere couche wire ou filaire qui correspond aux données sur le media de capture
- La 2eme couche est *Ethernet II* qui correspond à la couche de liaison de données du modèle OSI
- La 3eme couche est *Address Resolution Protocol*, un protocole effectuant la traduction d'une adresse IP à une adresse MAC.

Pour la trame *ICMP*, on distingue les 4 couches d'encapsulation suivantes

- La 1ère couche wire ou filaire qui correspond aux données sur le media de capture
- La 2ème couche est *Ethernet II* qui correspond à la couche de liaison de donnée de modèle OSI
- La 3ème couche est *Internet Protocol version 4* qui correspond à un relatif à la couche réseau.
- La 4ème couche est *Internet Control Message protocol* qui permet de communiquer des messages de contrôle et d'erreur de protocole.

Les différences entre ARP et ICMP sont les couches qui correspondent à *Internet Protocol version 4* et *Internet Control Message protocol*.