

TP 3

Mise en place de la configuration

Après avoir tout branché sur marionnet, on configure les machines m1, m2, m3 et m4 avec les adresses suivantes :

- m1 : 192.168.1.1
- m2 : 192.168.1.2
- m3 : 192.168.1.254 sur LAN1 et 10.10.0.254 sur LAN2

Voici les commandes utilisées pour configurer les machines :

- machine m1 :

```
ifconfig eth0 192.168.1.1 netmask 255.255.255.0
```

- machine m2 :

```
ifconfig eth0 192.168.1.2 netmask 255.255.255.0
```

- machine m3 :

Cette machine étant connectée aux deux réseaux, nous sommes obligés de lui attribuer deux adresses

```
ifconfig eth0 192.168.1.254 netmask 255.255.255.0  
ifconfig eth1 10.10.255.254 netmask 255.255.0.0
```

- machine m4 :

```
ifconfig eth0 10.10.0.4 netmask 255.255.0.0
```

Configuration d'une connexion avec la passerelle pour chaque machine

Pour établir une connexion entre deux machines appartenant à des réseaux différents, on doit configurer la passerelle sur la machine 3 car c'est la seule machine qui est reliée à la fois à LAN1 et à LAN2.

On utilise la commande *route* pour définir une route avec un autre réseau.

- sur les machines m1 et m2, on tape la commande suivante:

```
route add -net 10.10.0.0/16 gw 192.168.1.254
```

- sur la machine m3, on n'as pas besoin cette commande, car elle est déjà connectée aux deux réseaux, mais on doit cependant activer l'ip forwarding avec la commande suivante :

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

- sur la machine 4 on tape la commande

```
route add -net 192.168.1.0/24 gw 10.10.255.254
```

Test de la connexion

De la machine 1 vers la machine 4 :

```
ping 10.10.04
PING 10.10.04 (10.10.0.4) 56(84) bytes of data.
64 bytes from 10.10.0.4: icmp_seq=1 ttl=63 time=41.1 ms
64 bytes from 10.10.0.4: icmp_seq=2 ttl=63 time=1.46 ms
64 bytes from 10.10.0.4: icmp_seq=3 ttl=63 time=1.18 ms

--- 10.10.04 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2014ms
rtt min/avg/max/mdev = 1.181/14.583/41.102/18.752 ms
```

On constate que la passerelle est bien configurée.

Configuration de passerelle par défaut

On supprime les routes précédemment rajoutées et on configure m3 en tant que passerelle par défaut :

```
m1:~# route
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
192.168.1.0      *                255.255.255.0    U        0      0        0 eth0
10.10.0.0        192.168.1.254   255.255.0.0      UG        0      0        0 eth0

m1:~# route del -net 10.10.0.0/16 gw 192.168.1.254

m1:~# route add default gw 192.168.1.254

m1:~# route
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
192.168.1.0      *                255.255.255.0    U        0      0        0 eth0
default          192.168.1.254   0.0.0.0          UG        0      0        0 eth0
```

La différence en utilisant la configuration par défaut est que la machine m3 fait maintenant office de passerelle pour tous les réseaux et plus seulement entre les réseaux LAN1 et LAN2 que l'on a créé.

De plus, la commande route met beaucoup plus de temps à afficher la ligne correspondant à la passerelle lorsqu'elle est configurée par défaut.

Pour observer cette différence, il faudrait chercher à joindre un autre réseau et utiliser un outil comme wireshark pour analyser les trames.

Le rôle de ICMP dans le routage IP : un cas de figure

En tentant un ping sur la machine m4, on obtient le résultat indiqué dans le sujet (`connect: Network is unreachable`).

Configuration de l'IP de l'interface eth0 sur m1

Rien ne se passe car la passerelle n'est pas configurée sur m1.

Configuration de m3 comme passerelle par défaut sur m1

On observe des requêtes du protocole ARP, demandant quelle est la machine correspondant à l'adresse IP de la passerelle. Comme celle-ci n'est pas configurée, les requêtes restent sans réponse. Voici la capture ce qui passe dans le terminal de m2 :

```
11:33:53.640879 arp who-has 192.168.1.254 tell 192.168.1.1
11:33:54.658311 arp who-has 192.168.1.254 tell 192.168.1.1
11:33:55.665117 arp who-has 192.168.1.254 tell 192.168.1.1
11:33:56.663432 arp who-has 192.168.1.254 tell 192.168.1.1
11:33:57.686178 arp who-has 192.168.1.254 tell 192.168.1.1
11:33:58.694371 arp who-has 192.168.1.254 tell 192.168.1.1
11:33:59.693046 arp who-has 192.168.1.254 tell 192.168.1.1
11:34:00.718539 arp who-has 192.168.1.254 tell 192.168.1.1
11:34:01.728045 arp who-has 192.168.1.254 tell 192.168.1.1
11:34:02.719908 arp who-has 192.168.1.254 tell 192.168.1.1
```

Configuration des IPs des interfaces eth0 et eth1 sur m3

Une fois machine 3 configurée, on observe maintenant des requêtes du protocole ICMP envoyé par m1, mais comme l'ip forwarding n'est pas activé, celles-ci restent sans réponse. La message obtenu dans la section précédente est transformé en les messages suivantes:

```
11:37:22.806215 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 16906, seq 1, length 64
11:37:23.789535 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 16906, seq 2, length 64
11:37:24.802684 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 16906, seq 3, length 64
11:37:25.823218 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 16906, seq 4, length 64
11:37:26.835120 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 16906, seq 5, length 64
11:37:27.843369 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 16906, seq 6, length 64
11:37:28.846024 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 16906, seq 7, length 64
11:37:29.860548 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 16906, seq 8, length 64
```

Activation du routage sur m3

Une fois l'ip forwarding activé, on voit que m3 réponds à m1 en disant que m4 n'est pas atteignable en utilisant le protocole ICMP. Il ne s'agit ni d'un message de type REPLY ni d'un message de type ECHO mais d'un message de type HOST UNREACHABLE. Sur m4, on voit une requête ARP originant de m3, demandant à quelle machine correspond l'adresse 10.10.0.4.

```
11:42:01.187434 arp who-has 10.10.0.4 tell 10.10.255.254
11:42:02.174894 IP 192.168.1.1 > 10.10.0.4: ICMP echo request,id 21258,seq 12, length 64
11:42:02.189652 arp who-has 10.10.0.4 tell 10.10.255.254
11:42:03.189301 IP 192.168.1.1 > 10.10.0.4: ICMP echo request,id 21258,seq 13, length 64
11:42:03.195933 arp who-has 10.10.0.4 tell 10.10.255.254
11:42:04.196337 IP 192.168.1.1 > 10.10.0.4: ICMP echo request,id 21258,seq 14, length 64
11:42:04.216051 arp who-has 10.10.0.4 tell 10.10.255.254
11:42:05.212364 IP 192.168.1.1 > 10.10.0.4: ICMP echo request,id 21258,seq 15, length 64
```

Configuration de l'interface eth0 sur m4

La situation est améliorée dans le sens que m1 envoie des messages à m4(ICMP de type ECHO). Cependant, comme la passerelle n'est pas configurée sur m4, m4 ne réponds pas. Sur les deux réseaux, on observe des trames du protocole ICMP de type ECHO.

- capture du terminal sur la machine m2

```
11:45:08.597650 arp who-has 10.10.0.4 tell 10.10.255.254
11:45:09.591208 IP 192.168.1.1 > 10.10.0.4: ICMP echo request,id 24842,seq 11, length 64
11:45:09.596912 arp who-has 10.10.0.4 tell 10.10.255.254
11:45:10.594017 arp who-has 10.10.0.4 tell 10.10.255.254
11:45:10.598603 IP 192.168.1.1 > 10.10.0.4: ICMP echo request,id 24842,seq 12, length 64
11:45:11.605326 IP 192.168.1.1 > 10.10.0.4: ICMP echo request,id 24842,seq 13, length 64
11:45:11.629061 arp who-has 10.10.0.4 tell 10.10.255.254
11:45:12.621391 IP 192.168.1.1 > 10.10.0.4: ICMP echo request,id 24842,seq 14, length 64
11:45:12.636449 arp who-has 10.10.0.4 tell 10.10.255.254
11:45:13.628904 IP 192.168.1.1 > 10.10.0.4: ICMP echo request,id 24842,seq 15, length 64
```

- capture du terminal sur la machine m4

```
11:58:07.293833 IP 192.168.1.1 > 10.10.0.4: ICMP echo request,id 11530,seq 70,length 64
11:58:08.306231 IP 192.168.1.1 > 10.10.0.4: ICMP echo request,id 11530,seq 71,length 64
11:58:09.311346 IP 192.168.1.1 > 10.10.0.4: ICMP echo request,id 11530,seq 72,length 64
11:58:10.319305 IP 192.168.1.1 > 10.10.0.4: ICMP echo request,id 11530,seq 73,length 64
11:58:11.342406 IP 192.168.1.1 > 10.10.0.4: ICMP echo request,id 11530,seq 74,length 64
11:58:12.340854 IP 192.168.1.1 > 10.10.0.4: ICMP echo request,id 11530,seq 75,length 64
```

Configuration de m3 comme passerelle par défaut sur m4

Une fois la passerelle configurée, le ping **m1** → **m4** fonctionne finalement.

- Capture du terminal de m1:

```
64 bytes from 10.10.0.4: icmp_seq=75 ttl=63 time=57.3 ms
64 bytes from 10.10.0.4: icmp_seq=76 ttl=63 time=12.2 ms
64 bytes from 10.10.0.4: icmp_seq=77 ttl=63 time=15.2 ms
64 bytes from 10.10.0.4: icmp_seq=78 ttl=63 time=13.7 ms
64 bytes from 10.10.0.4: icmp_seq=79 ttl=63 time=11.1 ms
64 bytes from 10.10.0.4: icmp_seq=80 ttl=63 time=9.39 ms
64 bytes from 10.10.0.4: icmp_seq=81 ttl=63 time=15.8 ms
64 bytes from 10.10.0.4: icmp_seq=82 ttl=63 time=8.69 ms
64 bytes from 10.10.0.4: icmp_seq=83 ttl=63 time=9.74 ms
64 bytes from 10.10.0.4: icmp_seq=84 ttl=63 time=6.97 ms
64 bytes from 10.10.0.4: icmp_seq=85 ttl=63 time=11.5 ms
64 bytes from 10.10.0.4: icmp_seq=86 ttl=63 time=11.8 ms
64 bytes from 10.10.0.4: icmp_seq=87 ttl=63 time=13.1 ms
```

- capture du terminal de m2:

```
11:58:13.345546 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 11530, seq 76, length 64
11:58:13.356981 IP 10.10.0.4 > 192.168.1.1: ICMP echo reply, id 11530, seq 76, length 64
11:58:14.359249 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 11530, seq 77, length 64
11:58:14.367843 IP 10.10.0.4 > 192.168.1.1: ICMP echo reply, id 11530, seq 77, length 64
11:58:15.369658 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 11530, seq 78, length 64
11:58:15.379851 IP 10.10.0.4 > 192.168.1.1: ICMP echo reply, id 11530, seq 78, length 64
11:58:16.381012 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 11530, seq 79, length 64
11:58:16.390085 IP 10.10.0.4 > 192.168.1.1: ICMP echo reply, id 11530, seq 79, length 64
11:58:17.389202 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 11530, seq 80, length 64
11:58:17.398260 IP 10.10.0.4 > 192.168.1.1: ICMP echo reply, id 11530, seq 80, length 64
11:58:18.388151 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 11530, seq 81, length 64
11:58:18.397914 IP 10.10.0.4 > 192.168.1.1: ICMP echo reply, id 11530, seq 81, length 64
11:58:19.401059 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 11530, seq 82, length 64
11:58:19.408622 IP 10.10.0.4 > 192.168.1.1: ICMP echo reply, id 11530, seq 82, length 64
11:58:20.400344 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 11530, seq 83, length 64
11:58:20.407960 IP 10.10.0.4 > 192.168.1.1: ICMP echo reply, id 11530, seq 83, length 64
```

- capture du terminal de m4

```
11:58:12.370193 IP 10.10.0.4 > 192.168.1.1: ICMP echo reply, id 11530, seq 75, length 64
11:58:13.355082 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 11530, seq 76, length 64
11:58:13.355157 IP 10.10.0.4 > 192.168.1.1: ICMP echo reply, id 11530, seq 76, length 64
11:58:14.363652 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 11530, seq 77, length 64
11:58:14.363975 IP 10.10.0.4 > 192.168.1.1: ICMP echo reply, id 11530, seq 77, length 64
11:58:15.377705 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 11530, seq 78, length 64
11:58:15.377770 IP 10.10.0.4 > 192.168.1.1: ICMP echo reply, id 11530, seq 78, length 64
11:58:16.388501 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 11530, seq 79, length 64
11:58:16.388569 IP 10.10.0.4 > 192.168.1.1: ICMP echo reply, id 11530, seq 79, length 64
11:58:17.390363 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 11530, seq 80, length 64
11:58:17.390425 IP 10.10.0.4 > 192.168.1.1: ICMP echo reply, id 11530, seq 80, length 64
11:58:18.388536 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 11530, seq 81, length 64
11:58:18.388593 IP 10.10.0.4 > 192.168.1.1: ICMP echo reply, id 11530, seq 81, length 64
11:58:19.403950 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 11530, seq 82, length 64
11:58:19.404020 IP 10.10.0.4 > 192.168.1.1: ICMP echo reply, id 11530, seq 82, length 64
11:58:20.400865 IP 192.168.1.1 > 10.10.0.4: ICMP echo request, id 11530, seq 83, length 64
11:58:20.400925 IP 10.10.0.4 > 192.168.1.1: ICMP echo reply, id 11530, seq 83, length 64
```