

Aix Marseille Université - Campus Luminy

UFR des Sciences

Rapport de TP

Master Informatique

Module Réseaux.

TP n°3 :

Routage Avancé.

Réalisé par :

ZEMMOURI Yasmine.

1. Construction et configuration des machines :

La configuration des machines s'est faite avec ansible en version statique, les fichiers *config.yml* ainsi que les *vagrantFile* sont joints avec ce rapport.

→ Vérification des configurations :

L'exécution des fichiers se fait avec la commande : ***sudo ansible-playbook -c local -v config.yml***

VM1 :

```
m1reseaux@VM1:~$ ip addr show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 08:00:27:aa:44:5c brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 172.16.2.131/28 brd 172.16.2.143 scope global noprefixroute eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::c87e:4ab3:1776:55a5/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
m1reseaux@VM1:~$ ip addr show eth2
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 08:00:27:9b:9e:41 brd ff:ff:ff:ff:ff:ff
    altname enp0s9
    inet 172.16.2.151/28 brd 172.16.2.159 scope global noprefixroute eth2
        valid_lft forever preferred_lft forever
    inet6 fe80::5019:3a4c:2889:7c9e/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
m1reseaux@VM1:~$
```

VM2 :

```
m1reseaux@VM2:~$ ip addr show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 08:00:27:75:e4:a2 brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 172.16.2.132/28 brd 172.16.2.143 scope global noprefixroute eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::86af:a3f:e5a1:5620/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
m1reseaux@VM2:~$ ip addr show eth2
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 08:00:27:9e:22:da brd ff:ff:ff:ff:ff:ff
    altname enp0s9
    inet 172.16.2.162/28 brd 172.16.2.175 scope global noprefixroute eth2
        valid_lft forever preferred_lft forever
    inet6 fe80::afd0:fbce:b0:eeef/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
m1reseaux@VM2:~$
```

VM3:

```
m1reseaux@VM3:~$ ip addr show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
   group default qlen 1000
   link/ether 08:00:27:a2:22:bd brd ff:ff:ff:ff:ff:ff
   altname enp0s8
   inet 172.16.2.163/28 brd 172.16.2.175 scope global noprefixroute eth1
       valid_lft forever preferred_lft forever
   inet6 fe80::9b8e:49c7:7613:88ea/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
m1reseaux@VM3:~$ ip addr show eth2
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
   group default qlen 1000
   link/ether 08:00:27:b4:9d:5b brd ff:ff:ff:ff:ff:ff
   altname enp0s9
   inet 172.16.2.183/28 brd 172.16.2.191 scope global noprefixroute eth2
       valid_lft forever preferred_lft forever
   inet6 fe80::e655:7e77:3bc5:eb10/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
m1reseaux@VM3:~$
```

VM1-6:

```
m1reseaux@VM1-6:~$ ip addr show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
   group default qlen 1000
   link/ether 08:00:27:30:6c:cb brd ff:ff:ff:ff:ff:ff
   altname enp0s8
   inet6 fc00:1234:1::16/64 scope global noprefixroute
       valid_lft forever preferred_lft forever
   inet6 fe80::ffa4:69c8:27ba:c6a8/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
m1reseaux@VM1-6:~$ ip addr show eth2
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
   group default qlen 1000
   link/ether 08:00:27:b4:3e:4a brd ff:ff:ff:ff:ff:ff
   altname enp0s9
   inet 172.16.2.156/28 brd 172.16.2.159 scope global noprefixroute eth2
       valid_lft forever preferred_lft forever
   inet6 fe80::9cd:629f:ff6a:29e8/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
m1reseaux@VM1-6:~$
```

VM2-6:

```
m1reseaux@VM2-6:~$ ip addr show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
   group default qlen 1000
   link/ether 08:00:27:dc:29:e0 brd ff:ff:ff:ff:ff:ff
   altname enp0s8
   inet6 fc00:1234:1::26/64 scope global noprefixroute
       valid_lft forever preferred_lft forever
   inet6 fe80::ea74:155b:69c6:51cc/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
m1reseaux@VM2-6:~$ ip addr show eth2
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
   group default qlen 1000
   link/ether 08:00:27:d4:f1:a1 brd ff:ff:ff:ff:ff:ff
   altname enp0s9
   inet6 fc00:1234:2::26/64 scope global noprefixroute
       valid_lft forever preferred_lft forever
   inet6 fe80::246c:7341:1070:9636/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
m1reseaux@VM2-6:~$
```

VM3-6 :

```
m1reseaux@VM3-6:~$ ip addr show eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state U
group default qlen 1000
    link/ether 08:00:27:1b:df:9e brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet6 fc00:1234:2::36/64 scope global noprefixroute
        valid_lft forever preferred_lft forever
    inet6 fe80::505a:d4bb:a07b:39bf/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
m1reseaux@VM3-6:~$ ip addr show eth2
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state U
group default qlen 1000
    link/ether 08:00:27:3d:55:b6 brd ff:ff:ff:ff:ff:ff
    altname enp0s9
    inet 172.16.2.186/28 brd 172.16.2.191 scope global noprefixroute eth2
        valid_lft forever preferred_lft forever
    inet6 fe80::a603:3ed9:2332:402c/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
m1reseaux@VM3-6:~$
```

→ Le ping entre les machines VM1, VM2, VM3, VM1-6 (LAN3) et VM3-6 (LAN4), en d'autres termes les adresses IPv4, et entre les machines VM1-6 (LAN1-6), VM2-6 et VM3-6 (LAN3-6), les adresses IPv6.

→ Le ping ne marche pas entre les adresse de type IPv4 et IPv6, par exemple entre VM2 et VM2-6, les deux protocoles sont incompatibles.

2. Utilisation :

- a. Remettre une passerelle par défaut et un serveur DNS :

```
m1reseaux@VM3-6:~$ sudo dhclient eth0
RTNETLINK answers: File exists
```

- b. Après l'exécution des commandes demandées, on vérifie que le serveur est bien démarré:

```
root@VM3-6:/home/m1reseaux# update-inetd --add "echo stream tcp6 nowait nobod
y internal"
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
    LANGUAGE = "fr_FR:",
    LC_ALL = (unset),
    LANG = "fr_FR.UTF-8"
are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
root@VM3-6:/home/m1reseaux# service inetutils-inetd start
root@VM3-6:/home/m1reseaux# service inetutils-inetd status
000 inetutils-inetd.service - LSB: GNU Network Utilities internet superserver
   Loaded: loaded (/etc/init.d/inetutils-inetd; generated)
   Active: active (exited) since Tue 2023-10-24 10:05:48 UTC; 1min 9s ago
     Docs: man:systemd-sysv-generator(8)
   Process: 2034 ExecStart=/etc/init.d/inetutils-inetd start (code=exited, >
    CPU: 14ms

Oct 24 10:05:48 VM3-6 systemd[1]: Starting LSB: GNU Network Utilities intern>
Oct 24 10:05:48 VM3-6 inetutils-inetd[2034]: Not starting internet superserv>
Oct 24 10:05:48 VM3-6 systemd[1]: Started LSB: GNU Network Utilities interne>
lines 1-10/10 (END)
```

- c. Après test du service, on observe les trames sur VM2-6 :

Capture en cours de eth1 (on VM2-6)

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
5	35.224190949	0.0.0.0	255.255.255.255	DHCP	325	DHCP Discover
6	58.615170560	fe80::ea74:155b:69c...	ff02::1	ICMPv6	110	Router Advertis
7	73.856745886	fe80::ea74:155b:69c...	ff02::1	ICMPv6	110	Router Advertis
8	77.462437739	0.0.0.0	255.255.255.255	DHCP	325	DHCP Discover
9	99.882039380	0.0.0.0	255.255.255.255	DHCP	325	DHCP Discover
10	102.002003586	fe80::ea74:155b:69c...	ff02::1	ICMPv6	110	Router Advertis

Frame 1: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface eth1, id 0

Ethernet II, Src: PcsCompu_dc:29:e0 (08:00:27:dc:29:e0), Dst: IPv6mcast_01 (33:33:00:00:00:01)

Internet Protocol Version 6, Src: fe80::ea74:155b:69c6:51cc, Dst: ff02::1

Internet Control Message Protocol v6

```

0000  33 33 00 00 00 01 08 00 27 dc 29 e0 86 dd 60 0f 33... ..)....
0010  49 70 00 38 3a ff fe 80 00 00 00 00 00 00 ea 74 Ip:8:... ..t
0020  15 5b 69 c6 51 cc ff 02 00 00 00 00 00 00 00 00 ..[i.Q.....
0030  00 00 00 00 00 01 86 00 d1 ef 40 00 00 5a 00 00 .....@.Z..
0040  00 00 00 00 00 00 03 04 40 c0 00 00 01 2c 00 00 .....@.....
0050  00 78 00 00 00 00 fc 00 12 34 00 01 00 00 00 00 ..x.....4....
0060  00 00 00 00 00 00 01 01 08 00 27 dc 29 e0 .....(').

```

- VM1-6 envoie d'abord une demande pour trouver l'adresse de VM3-6. Une fois qu'il reçoit la réponse, VM1-6 commence à échanger des informations avec VM3-6. VM2-6 assure que ces messages atteignent la bonne destination (VM3-6).

3. Serveur applicatifs IPv6 :

- a. La compilation des fichiers se fait via la commande : **javac -source 11 -target 11 EchoServer.java EchoClient.java**
 Puis, copier le bytecode dans le fichier partage : **cp *.class ../partage**
- b. Lancement du serveur sur VM3-6 :

```

m1reseaux@VM3-6:~$ cd /mnt/partage
m1reseaux@VM3-6:/mnt/partage$ java EchoServer 1234
Lancement du serveur sur le port 1234

```

- c. Lancement du client sur VM2-6 :

```
m1reseaux@VM1-6:~$ cd /mnt/partage
m1reseaux@VM1-6:/mnt/partage$ java EchoClient fc00:1234:2::36 1234
Essai de connexion ? fc00:1234:2::36 sur le port 1234

le n? de la socket est : Socket[addr=/fc00:1234:2:0:0:0:0:36,port=1234,localp
ort=57916]
re?u: Bonjour /fc00:1234:1:0:0:0:0:16! (vous utilisez le port 57916)
```

- d. Différence entre quitter avec Ctrl+C et Ctrl+D :

VM1-6 :

```
m1reseaux@VM1-6:/mnt/partage$ java EchoClient fc00:1234:2::36 1234
Essai de connexion ? fc00:1234:2::36 sur le port 1234

le n? de la socket est : Socket[addr=/fc00:1234:2:0:0:0:0:36,port=1234,localp
ort=57916]
re?u: Bonjour /fc00:1234:1:0:0:0:0:16! (vous utilisez le port 57916)
^Cm1reseaux@VM1-6:/mnt/partage$ java EchoClient fc00:1234:2::36 1234
Essai de connexion ? fc00:1234:2::36 sur le port 1234

le n? de la socket est : Socket[addr=/fc00:1234:2:0:0:0:0:36,port=1234,localp
ort=40646]
re?u: Bonjour /fc00:1234:1:0:0:0:0:16! (vous utilisez le port 40646)
Connexion termin?e !!
H?te distant inform?...
Connexion termin?e par l'h?te distant
Fin de la session.
m1reseaux@VM1-6:/mnt/partage$
```

VM3-6 :

```
m1reseaux@VM3-6:/mnt/partage$ java EchoServer 1234
Lancement du serveur sur le port 1234
[/fc00:1234:1:0:0:0:0:16:57916]: Termin?...
[/fc00:1234:1:0:0:0:0:16:40646]: Termin?...
```

→ Lorsqu'on appuie sur Ctrl-C, le client envoie un signal d'interruption au programme, ce qui peut entraîner une fermeture brusque sans que la connexion réseau soit correctement gérée. En revanche, l'utilisation de Ctrl-D envoie un signal de fin de fichier, ce qui est interprété comme la fin de l'entrée standard.

→ Ctrl-D est généralement préféré car il permet au client de fermer proprement la connexion réseau, effectuant ainsi des opérations de nettoyage adéquates

- e. Le serveur est écouté sur le port 1234. Cela est configurable en changeant le 2e argument de la commande 'java EchoServer **1234**' au numéro du port souhaité.

f. Les ports utilisés sur VM3-6 :

```
m1reseaux@VM3-6:~$ ss -plat6
State      Recv-Q      Send-Q      Local Address:Port      Peer Address:Port
Process
LISTEN     0            50          *:1234                  *:*
users:((("java",pid=2159,fd=5))
LISTEN     0           128          [::]:ssh                [::]:*
LISTEN     0           10          *:echo                  *:*
```

→ Chaque ligne présente des détails tels que l'état actuel du socket, la file d'attente de réception (Recv-Q) et la file d'attente d'envoi (Send-Q), ainsi que l'adresse locale et le port sur lesquels le socket est écouté.

g. Oui, il est possible de lancer un serveur en Java et un serveur en C en même temps. Les langages de programmation tels que Java et C sont conçus pour fonctionner sur des systèmes d'exploitation multitâches, ce qui signifie qu'ils peuvent exécuter plusieurs processus simultanément. Chaque serveur fonctionnera dans son propre processus et utilisera des ports différents pour écouter les connexions entrantes.

4. Autres clients :

Le telnet ne marche pas sur ma configuration sur VM1 :

```
m1reseaux@VM1:/mnt/partage$ telnet fc00:1234:2::36 1234
Trying fc00:1234:2::36...
telnet: Unable to connect to remote host: Network is unreachable
m1reseaux@VM1:/mnt/partage$ python3 echoclient.py fc00:1234:2::36 1234
le n° de la socket est : 3
Essai de connexion 00 fc00:1234:2::36 ( fc00:1234:2::36 ) sur le port 1234
Erreur Connexion : [Errno 101] Network is unreachable
m1reseaux@VM1:/mnt/partage$
```

→ J'ai une erreur de connexion : Network is unreachable, cette erreur n'est pas présente sur VM1-6.

→ Supposons que la connexion marche : cela illustre le paradigme client/serveur dans le contexte de la communication réseau. Les commandes `telnet`, `java EchoClient`, et `python echoclient.py` sont des exemples de clients qui se connectent à un serveur Echo distant sur l'adresse `fc00:1234:2::36` et le port `1234`. Chacun de ces clients, exécuté depuis VM1, envoie des données au serveur Echo (situé sur VM3-6) et attend des réponses. Le serveur Echo, conformément à ce paradigme, renvoie simplement les données reçues. Les clients varient, allant de l'outil Telnet standard aux clients personnalisés écrits en Java et Python, mais l'essence du paradigme client/serveur demeure la même.