TP 2
R301: IPv6

Sommaire

INTRODUCTION

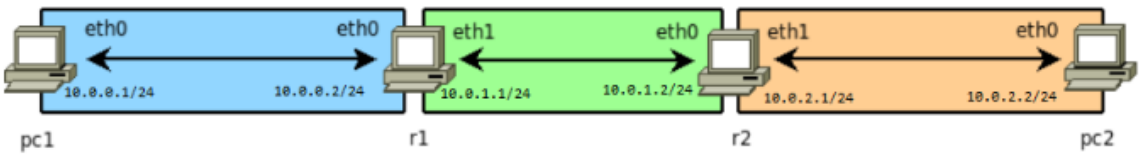
Face à la pénurie d'adresses IPv4 dans le monde, la solution proposée est d'utiliser un nouveau système d'adressage qui de disposer d'un nombre d'adresses beaucoup le plus important, le protocole IPv6.
Les adresses IPv6 sont codées sur 128 bits soit 16 octets, ce qui représente 2^ 128 adresses contre 2^32 en IPv4. La notation décimale pointée employée pour les adresses IPv4 (par exemple 172.31.128.1) est abandonnée au profit d'une écriture hexadécimale, où les 8 groupes de 2 octets (soit 16 bits par groupe) sont séparés par un signe deux-points ":" : 2001:0db8:0000:85a3:0000:0000:ac1f:8001.
La notion historique de classes a totalement disparu avec IPv6, au profit de l'utilisation exclusive des préfixes et de la notation CIDR avec le slash et le masque, déjà utilisés en IPv4.

OBJECTIF

L'objectif de notre TP est de mettre en place un réseau virtuel en utilisant le protocole IPv6 et d'explorer les aspects du routage IPv6. Pour ce faire, nous utiliserons une machine virtuelle Debian avec les logiciels Netkit, Wireshark, SSH, et IPv6calc.

Netkit, l'environnement choisi pour cette expérience, offre la possibilité de configurer et de réaliser des expériences de mise en réseau à faible coût et avec une facilité d'utilisation. Ces équipements de mise en réseau sont virtuels, mais ils reproduisent de nombreuses caractéristiques des équipements réels, y compris une interface de configuration.

La topologie du TP à déployer dans Netkit est la suivante :



A – Récupération de la VM

1) Récupérer le fichier debianIPv6.ova et importer l'image de la VM Debian VirtualBox

Premièrement, nous allons récupérer la VM sur le drive et on importe sur Oracle VirtualBox.

Identifiants :

- user = root pass = class
- user = userrt pass = userrt

B – Création du LAB dans NETKIT

1) Dans le dossier /LAB-RT créer un fichier "lab.conf", ajouter les informations ci-dessous et enregistrer le fichier

Nous avons créé un fichier lab.conf et nous allons ajouter les informations ci-dessous :

```
LAB_DESCRIPTION="Lab IPv6 Netkit"
LAB_AUTHOR="Etudiant RT"
pc1[0]=dca
r1[0]=dca
r1[1]=dcb
r2[0]=dcb
r2[1]=dcc
pc2[0]=dcc
```

Ce fichier "`lab.conf`" sert à décrire la configuration du laboratoire Netkit. Il spécifie les machines virtuelles (pc1,pc2, r1, r2) et leurs interfaces, indiquant comment elles sont connectées les unes aux autres.

Par la suite, nous avons créé les fichiers d'initialisation pour chaque machine virtuelle dans le même dossier (/LAB-RT). Pour pc1, un fichier "pc1.startup" a été créé contenant les informations suivantes :

Fichier "pc1.startup" pour pc1 :

```
ip addr add 10.0.0.1/24 dev eth0
ip link set eth0 up
```

Nous avons répété ce processus pour les autres VMs (r1, r2, et pc2), en veillant à ajuster les adresses des interfaces pour correspondre à la topologie spécifiée.

Toujours dans le dossier, nous avons créé un dossier pour chaque VM du laboratoire dans le dossier /LAB-RT. Par exemple, pour PC1, nous avons exécuté la commande suivante :

```
pc1:~# mkdir /LAB-RT/pc1
```

Nous avons attribué les droits nécessaires au dossier /LAB-RT pour permettre l'exécution des scripts d'initialisation avec la commande :

```
pc1:~# chmod -R 777 /LAB-RT
```

Enfin, nous avons démarré les VMs Netkit en utilisant la commande lstart suivie du nom de la machine. Pour démarrer toutes les VMs, nous avons utilisé l'option -f :

```
pc1:/LAB-RT# lstart pc1
```

Astuce : Pour démarrer toutes les VMs, nous avons utilisé l'option -f :

Cette option permet de lancer plusieurs machines virtuelles à la fois sans attendre que la machine précédente termine sa “phase de boot”

```
pc1:/LAB-RT# lstart -f
```

```
userrt@DebianIPv6:/LAB-RT$ lstart -f

======================== Starting lab ============================
Lab directory: /LAB-RT
Version:       <unknown>
Author:        Etudiant RT
Email:         <unknown>
Web:           <unknown>
Description:
Lab IPv6 Netkit
==================================================================
Starting "pc1"...
Starting "pc2"...
Starting "r1"...
Starting "r2"...

The lab has been started.
==================================================================
```

Cela a lancé le processus de démarrage pour toutes les machines virtuelles du laboratoire.

C – Révision IPv4

1. **Vérification de la Connectivité entre les Machines :**

Après le démarrage des machines, nous avons procédé à la vérification de la connectivité entre les différentes paires de machines, à savoir :

- Entre PC1 et r1
- Entre R1 et R2
- Entre R2 et PC2
- Entre PC1 et PC2

Observations : Nous avons constaté que la connectivité entre toutes les machines n'était pas établie. Des problèmes de communication étaient présents entre certaines paires de machines.

Actions Possibles pour Établir une Connectivité Complète : Pour garantir une communication réussie entre toutes les machines, plusieurs actions auraient pu être entreprises :

- **Vérification des configurations IP :** Nous allons examiner attentivement les configurations IP de chaque interface pour s'assurer qu'elles appartiennent au même sous-réseau.
- **Inspection des Tables de Routage :** Nous allons vérifier les tables de routage sur les routeurs pour assurer un routage approprié entre les différents sous-réseaux.
- **Dépannage des Connexions :** Nous allons identifier et résoudre tout problème de connectivité, comme des erreurs de câblage virtuel ou des configurations incorrectes.
- **Tests de Connectivité :** Nous allons **u**tiliser des commandes telles que `ping` pour diagnostiquer spécifiquement les problèmes de connectivité entre les machines.

D – IPv6 : Prise en main

```
pc1 login; root (automatic login)
Last login: Fri Nov 10 09:13:30 UTC 2023 on tty1
pc1:# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
inet6 :: 1/128 scope host
valid_lft forever preferred_lft forever
2: teq10: <NOARP> mtu 1500 qdisc noop qlen 100
link/void
3: eth0: <BROADCAST, MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
pfifo_fast qlen 1000
link/ether 6e:5f:98:37:0c:07 brd ff:ff:ff:ff:ff:ff
inet 10.0.0.1/24 scope global eth0
inet6 fe80 :: 6c5f:98ff:fe37:c07/64 scope link
valid_lft forever preferred_lft forever
pc1:^# ip -6 addr 1s dev lo
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436
inet6 :: 1/128 scope host
valid_lft forever preferred_lft forever
```

En exécutant ces commandes sur pc1, nous obtiendrons les informations nécessaires sur la configuration des interfaces réseau IPv4 et l'adresse IPv6 de l'interface lo.

Pour envoyer 3 paquets ICMP Echo Request vers l'interface lo en IPv6, nous avons utilisé la commande suivante :

```
pc1:~# ping6 -c 3 ::1
```

```
pc1:~# ping6 -c 3 ::1
PING ::1(::1) 56 data bytes
64 bytes from ::1: icmp_seq=1 ttl=64 time=0.097 ms
64 bytes from ::1: icmp_seq=2 ttl=64 time=0.041 ms
64 bytes from ::1: icmp_seq=3 ttl=64 time=0.043 ms
```

l'Adresse:1 représente l'adresse IPv6 de la boucle locale associée à l'interface lo.

Afin de déterminer le(s) nom(s) associé(s) à l'adresse l0 en IPv6, nous avons consulté le contenu du fichier /etc/hosts en utilisant la commande :

```
pc1:~# cat /etc/hosts
```

```
pc1:~# cat /etc/hosts
127.0.0.1 pc1
127.0.0.1 localhost localhost.localdomain

::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
```

Le fichier /etc/hosts contient des associations entre noms et adresses IP, y compris celles de la boucle locale en IPv6.

Par la suite, pour Pour relever la configuration IPv6 de l'interface eth0, nous avons utilisé la commande :

```
pc1:~# ip -6 addr ls dev eth0
```

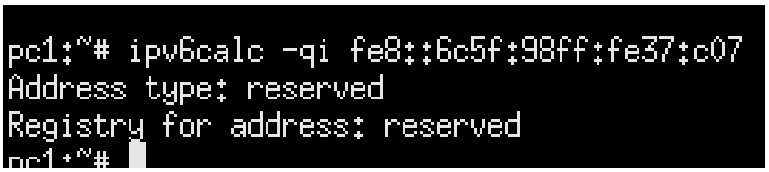
```
pc1:~# ip -6 addr ls dev eth0
3: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qlen 1000
    inet6 fe80::6c5f:98ff:fe37:c07/64 scope link
       valid_lft forever preferred_lft forever
```

Cette commande nous a fourni des détails sur les adresses IPv6 associées à l'interface eth0.

Pour déterminer le type d'adresse affectée à l'interface eth0, deux approches ont été utilisées. Tout d'abord, en utilisant la commande ipv6calc :

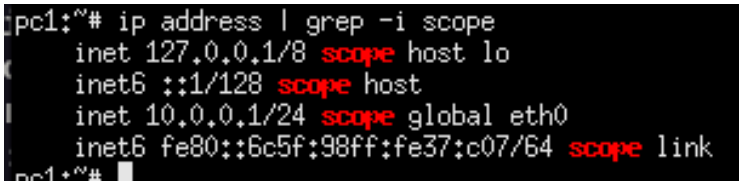
```
pc1:~# ipv6calc -qi fe8::6c5f:98ff:fe37:c07
```

```
pc1:~# ipv6calc -qi fe8::6c5f:98ff:fe37:c07
```



De plus, en utilisant la commande "ip address" avec un filtrage sur le scope :

```
pc1:~# ip address | grep -i scope
```



Ces commandes ont permis d'identifier le type d'adresse IPv6 (tel que global ou link-local) associé à l'interface eth0.

Calcul de l'Adresse "Link-Local" IPv6 à partir de l'Adresse MAC :

Pour déterminer l'adresse "link-local" IPv6 associée à une machine dont l'adresse MAC est "**00:4f:4e:08:25:1a**", nous avons suivi le processus standard d'attribution d'une adresse "link-local" basée sur l'adresse MAC en IPv6.

Conversion de l'Adresse MAC en Binaire :
L'adresse MAC "00:4f:4e:08:25:1a" a été convertie en binaire comme suit : **000011110100111100100110000010000001000010010110000001**.

Insertion du Préfixe "fe80::/64" :
Le préfixe "**fe80::/64**" a été ajouté à la partie basse de l'adresse MAC en binaire. Le résultat est l'adresse "link-local" IPv6 : **fe80::24f:4eff:fe08:251a**.

Ainsi, l'adresse "link-local" IPv6 associée à la machine ayant l'adresse MAC "**00:4f:4e:08:25:1a**" est **fe80::24f:4eff:fe08:251a.**

E – IPv6 : paramétrage IP Statique de base

Nous avons ajouté une adresse IPv6 à l'interface eth0 du poste PC1 en utilisant la commande suivante :

```
pc1:~# ip addr add 2001:db8:46::1/64 dev eth0
```

Pour vérifier la configuration, nous avons affiché les informations :

```
pc1:~# ip address
```

On voit que l'adresse ajoutée, 2001:db8:46::1/64, est une adresse unicast IPv6 a bien été attribuée sur le l'interface eth0.

Affichage de la table de routage IPv6 : Pour visualiser la table de routage IPv6 du poste pc1, nous avons utilisé la commande :

```
pc1:~# route -n -A inet6
```

```
pc1:~# route -n -A inet6
Kernel IPv6 routing table
Destination                    Next Hop                   Flag Met Ref Use If
2001:db8:46::/64               ::                         U    256 0     0 eth0
fe80::/64                      ::                         U    256 0     0 eth0
::/0                           ::                         !n   -1  1     1 lo
::1/128                        ::                         Un   0   1     0 lo
2001:db8:46::1/128             ::                         Un   0   1     0 lo
fe80::6c5f:98ff:fe37:c07/128   ::                         Un   0   1     0 lo
ff00::/8                       ::                         U    256 0     0 eth0
::/0                           ::                         !n   -1  1     1 lo
```

Les options `-n` permettent d'afficher les adresses en numérique au lieu du nom, et `-A inet6` spécifie la famille d'adresse IPv6.

Démarrage du service SSH et observation des connexions réseau IPv6 : Nous avons activé le service SSH sur pc1 et observé les connexions réseau associées à la pile IPv6 en utilisant les commandes :

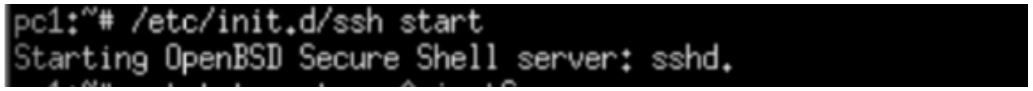
Avant le Démarrage du Service SSH :

```
pc1:~# netstat -natup -A inet6
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
```

```
pc1:"# netstat -natup -A inet6
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State
PID/Program name
```

La sortie indique qu'aucune connexion réseau TCP (IPv6) n'est actuellement établie avant le démarrage du service SSH.

Démarrage du Service SSH :




```
pc1:"# /etc/init.d/ssh start
Starting OpenBSD Secure Shell server: sshd.
```

Nous avons donc mis en marche ssh.

Après le Démarrage du Service SSH :


```
pc1:"# netstat -natup -A inet6
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State
PID/Program name
tcp6 0 0 :::22  0 :::* Listen 528/sshd
```

La sortie indique que le service SSH est activé et en écoute sur le port 22 en IPv6, prêt à accepter les connexions entrantes. La machine pc1 est maintenant accessible via SSH sur IPv6.

Catégories d'Usages des Adresses Unicasts : Les adresses unicast IPv6 peuvent être catégorisées en deux usages distincts :

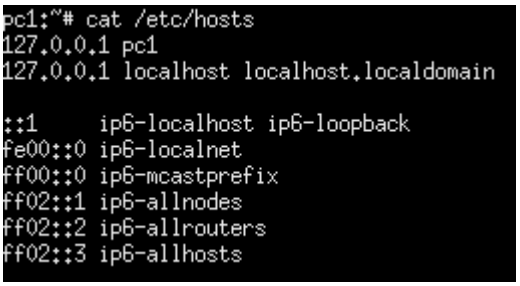
- **Globales :** Ces adresses sont utilisées sur l'ensemble de l'Internet et permettent une communication globale.
- **Lien-locals :** Limitées à la communication sur le même lien local (sous-réseau), ces adresses sont utilisées pour les communications internes à un réseau restreint.

F – IPv6 : Adresse Multicast

Nom	Adresse	Équivalent IPv4	Fonction
all-nodes	ff02::1	224.0.0.1	Tous les nœuds et routeurs du lien local. Utilisée, par exemple, pour les interfaces sans adresse qui souhaitent recevoir une réponse.
all-routers	ff02::2	224.0.0.2	Tous les routeurs du lien local. Utilisée, par exemple, pour solliciter une annonce de préfixe sur le réseau.

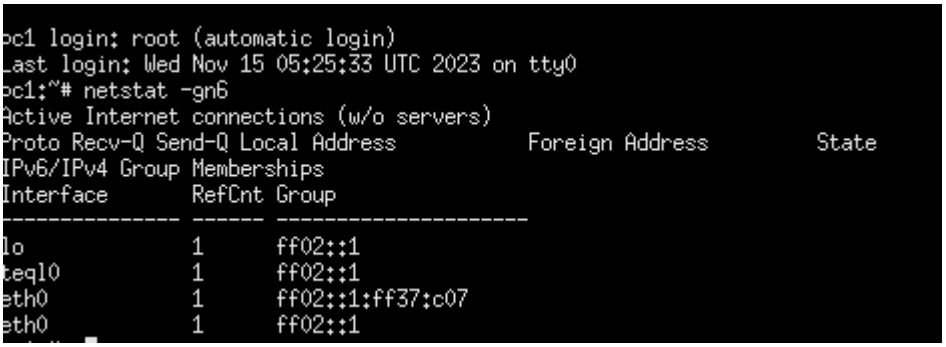
1) Contenu du fichier /etc/hosts et adresses multicast : Nous avons affiché le contenu du fichier /etc/hosts en utilisant la commande suivante :

```
pc1:~# cat /etc/hosts
```



2) Abonnements multicast pour le poste pc1 : Nous avons affiché les abonnements multicast pour le poste pc1 en utilisant deux commandes

```
pc1:~# netstat -g6n
```



Cette commande affiche les groupes multicast auxquels la machine est abonnée, indiquant les adresses multicast associées.

```
pc1:~# ip -6 maddress show dev eth0
```

Cette commande, spécifique à la gestion des adresses multicast, montre les adresses multicast associées à l'interface eth0, révélant les abonnements multicast spécifiques à cette interface.

G – IPv6 : Le protocole

1) Configuration adresse IPV6

Nous ajoutons sur la machine R1 l'adresse IPV6 **2001:db8:46::2/64** via la commande suivante:

```
r1:~# ip addr add 2001:db8:46::2/64 dev eth0
```

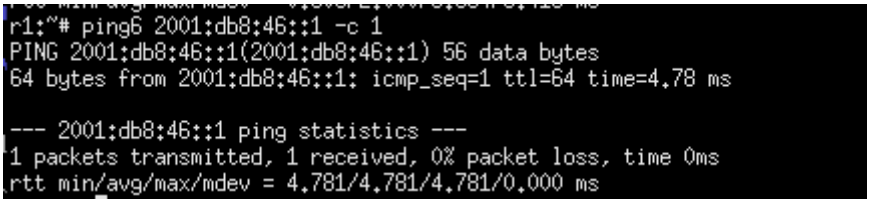
Une fois l'adresse IP ajoutée, nous vérifions la présence de l'adresse IP avec la commande suivante

```
r1:~# ip addr
```

Nous pouvons voir la présence l'adresse IPV6 que nous venons d'ajouter

Nous allons maintenant émettre 1 paquet ICMPV6 afin de tester la connectivité entre les machines R1 et PC1, pour cela, nous allons utiliser la commande suivante depuis

```
r1:~# ping6 2001:db8:46::1 -c 1
```



Nous pouvons voir que les machines peuvent communiquer

2) Capture de Trame avec tcpdump sur pc1 et Analyse avec Wireshark :

Par la suite sur le pc1, nous avons réalisé une capture de trame avec tcpdump pendant le ping de r1 vers pc1 :

```
pc1:~# tcpdump -i eth0 ip6 -w /hosthome/r1pc1.pcap
```

3) Capture et analyse Wireshark :

Après l'arrêt de la capture avec CTRL+C, nous avons analysé le fichier de capture avec Wireshark sur le VM Virtualbox.

5) Analyse du fichier “r1pc1.pcap” avec wireshark

a) Le champ type des trames Ethernet_II a comme valeur “IPV6” ce champ sert à indiquer le protocole de la couche supérieur

b) **Version des Paquets IP :** La valeur du champ "Version" des paquets IP était systématiquement 6, indiquant qu'il s'agissait de paquets IPv6.

c) **Valeur du champ Next header des paquets IP :** Le champ Next header des paquets IPv6 à la valeur 58, qui correspond au protocole ICMPv6. La valeur a été vérifiée dans le fichier /etc/protocols.

d) **Valeur du champ Payload length des paquets IP :** Le champ Payload length indique la longueur du chargement utile (payload) en octets. Il varie en fonction du contenu du paquet.

e) **Type et code des messages ICMP envoyés et reçus :** Les messages ICMP échangés sont des requêtes et réponses Echo (ping) avec les codes correspondants.

f) **Trames ARP et adresses de diffusion générale :** Dans IPv6, les trames ARP n'existent plus. Les adresses de diffusion générale sont remplacées par les adresses multicast, notamment pour la découverte des voisins.

g) **Valeur utilisée comme adresse destination dans la trame Ethernet_II :** L'adresse de destination dans la trame Ethernet_II est l'adresse MAC de la machine pc1.

h) **Rôle des paquets "icmpv6 neighbor solicitation" :** Ces paquets sont des requêtes de découverte des voisins dans IPv6, utilisés pour obtenir l'adresse MAC d'une machine associée à une adresse IPv6.

6) Consultation du cache entre les voisins sur PC1 et R1 :

```
pc1:# ip neigh show
```

H – Routage Statique IPv6

Configuration des Adresses IPv6 Statiques pour les Périphériques :

```
Pour pc1 :
pc1:~# ip addr add 2001:db8:46::1/64 dev eth0

Pour pc2 :
pc2:~# ip addr add 2001:db8:64:1::1/64 dev eth0

Pour r1 :
r1:~# ip addr add 2001:db8:46::2/64 dev eth0
r1:~# ip addr add 2001:db8:64::1/64 dev eth1

Pour r2 :
r2:~# ip addr add 2001:db8:64::2/64 dev eth0
r2:~# ip addr add 2001:db8:64:1::2/64 dev eth1
```

Configuration de PC1 :

a. Vérification de l'interface eth0 :

```
pc1:# ip link set dev eth0 up
```

b. Affectation d'une adresse IPv6 à eth0 :

```
pc1:# ip addr add 2001:db8:46::1/64 dev eth0
```

c. Affichage de la table de routage de pc1 :

```
pc1:# route -n -A inet6
```

Configuration des routes statiques :

PC1 :

```
pc1:~# ip -6 route add 2001:db8:64:1::/64 via 2001:db8:46::2 dev
eth0
```

PC2 :

```
pc2:~# ip -6 route add 2001:db8:46::/64 via 2001:db8:64:1::2 dev
eth0
```

R1 :

```
r1:~# ip -6 route add 2001:db8:64:1::/64 via 2001:db8:64::2 dev
eth1
r1:~# echo "1" > /proc/sys/net/ipv6/conf/default/forwarding
r1:~# echo "1" > /proc/sys/net/ipv6/conf/all/forwarding
```

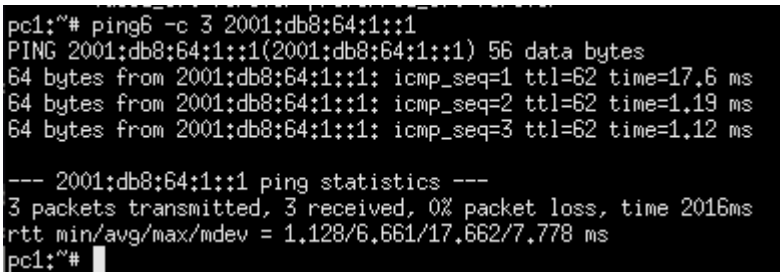
R2 :

```
r2:~# ip -6 route add 2001:db8:46::/64 via 2001:db8:64::1 dev eth0
r2:~# echo "1" > /proc/sys/net/ipv6/conf/default/forwarding
r2:~# echo "1" > /proc/sys/net/ipv6/conf/all/forwarding
```

Vérification de la Connectivité :

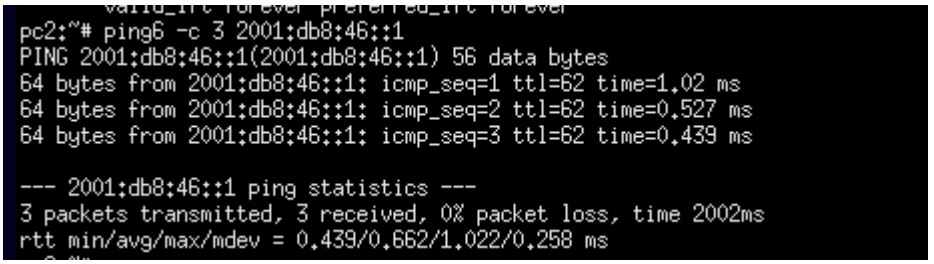
De pc1 vers pc2 :

```
pc1:~# ping6 -c 3 2001:db8:64:1::1
```



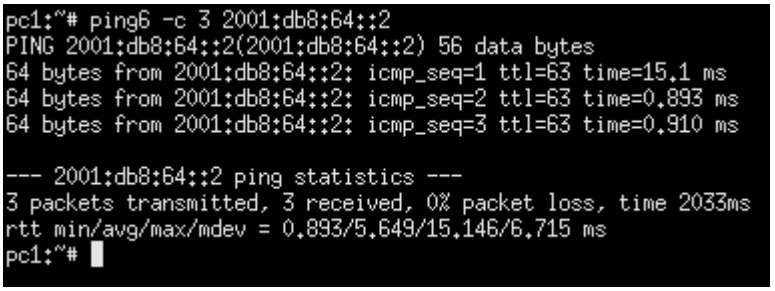
De pc2 vers pc1 :

```
pc2:~# ping6 -c 3 2001:db8:46::1
```



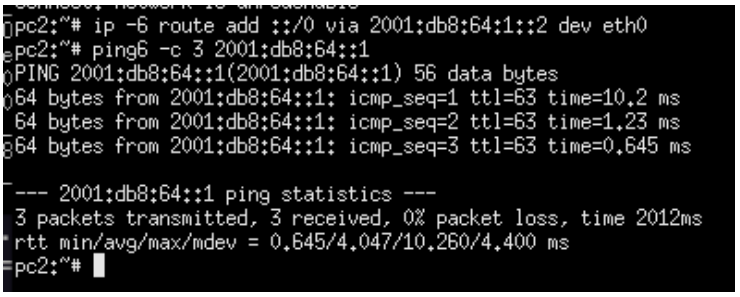
De pc1 vers r2 :

```
pc1:~# ping6 -c 3 2001:db8:64::2
```



De pc2 vers r1 :

```
pc2:~# ping6 -c 3 2001:db8:64::1
```

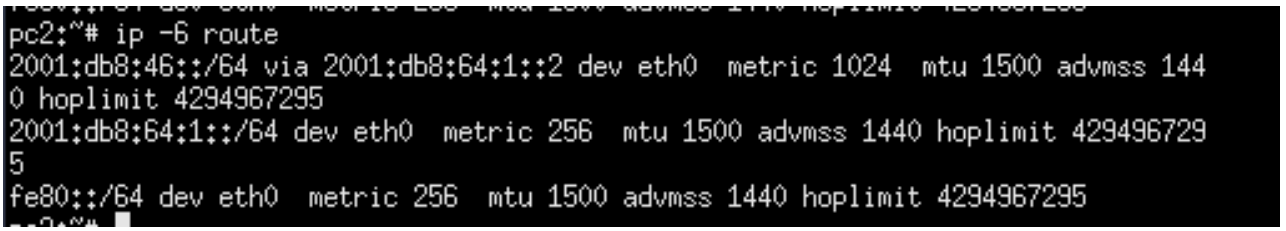


Nous pouvons voir que les machines peuvent communiquer entre elles

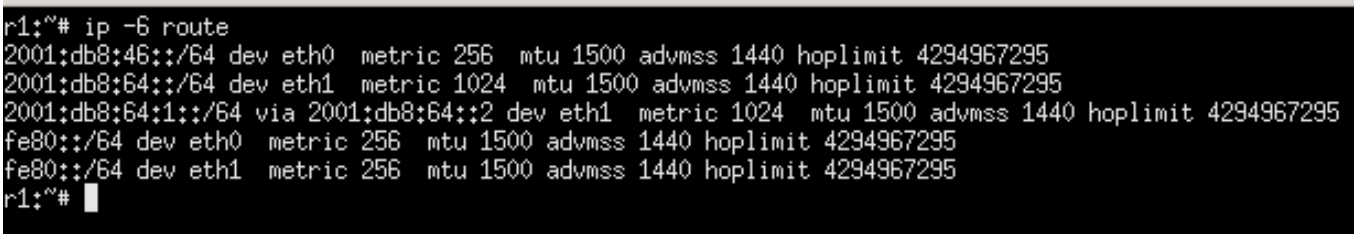
Tables de routages

PC1:

PC2:



R1:



R2:

Les machines ont toutes connaissance les unes des autres comme le montre les table de routages ci-dessus

Conclusion

En conclusion, ce TP sur l'IPv6 a permis d'explorer de manière approfondie la mise en place d'un réseau virtuel utilisant cette nouvelle génération d'adresses IP.

L'introduction a souligné la nécessité du passage à IPv6 en raison de la pénurie d'adresses IPv4. Nous avons ensuite détaillé les étapes pour la création du laboratoire dans Netkit, et la configuration des machines virtuelles.

La section sur IPv6 a couvert la prise en main, le paramétrage IP statique de base, l'utilisation d'adresses multicast, et l'exploration du protocole IPv6. Nous avons également examiné la capture et l'analyse de trames avec Wireshark, illustrant les spécificités du protocole IPv6.

Enfin, la configuration du routage statique IPv6 a été détaillée pour chaque machine, avec une vérification de la connectivité entre les différents éléments du réseau.

Ce TP a fourni une expérience pratique approfondie dans la mise en œuvre du protocole IPv6, mettant en lumière ses avantages et ses spécificités par rapport à IPv4. Les étapes détaillées, les commandes utilisées et les résultats obtenus ont contribué à renforcer la compréhension des concepts liés à IPv6 et à la configuration d'un réseau IPv6.