

# TP4 compte rendu SERRANO/NAIDJA

## 1.2 Client IPv6

1) l'adresse commune à toutes les stations est ::1

2) En utilisant un client netcat, la connexion ne marche pas car il ne sait pas convertir l'ipv4 en ipv6. En utilisant un client java cela marche car sait transformer l'ipv4 en ipv6.

## 1.3 Serveur Ipv6

3) La connexion est établie avec nc -6 ::1 1234 car le serveur est compatible avec autant ipv4 que ipv6. Il semble que Java fasse la gestion des adresses.

## 1.4

fe80::6600:6aff:fe5a:ec5a

## 2.1 Definition

La double pile IP est la duplication de toutes les couches de protocoles. Cela permet d'utiliser les mêmes protocoles de transport pour IPv4 et IPv6, les 2 types d'adresse IP sont gérés.

## 2.2

Si on lui donne directement une adresse, le client est capable de voir la différence entre une adresse IPv4 et IPv6 et fait donc le choix en analysant l'adresse.

Si on lui donne un nom de domaine, le client fait en fonction de la préférence du serveur.

## 2.3

Notre serveur écoute sur :

localhost:domain 0.0.0.0.\*

localhost:8118 0.0.0.0.\*

localhost:ipp 0.0.0.0.\*

```
localhost:9050 0.0.0.0.*  
ip6-localhost:ipp [::]:*
```

Ce qui semble être uniquement l'adresse localhost.

Pour qu'un serveur écoute uniquement sur une unique adresse, il faut

.preferIPv4Stack restreint la communication en empêchant de communiquer avec des hôtes qui utilisent uniquement des adresses IPv6. Cela peut être utile pour une application utilisant uniquement des sockets IPv4.

.preferIPv6Addresses change la préférence de base, qui est de favoriser une communication via IPv4, pour favoriser une communication par adresses IPv6 si c'est possible.

Pour configurer un serveur uniquement en IPv4, il faut mettre la configuration `java.net.preferIPv4stack` à 'true'.

Pour configurer un serveur uniquement en IPv6,

Avoir un serveur simple peut être utile pour une application qui ne peut gérer qu'un seul type d'adresse par exemple.

2.5

Windows favorise les connexions avec le protocole IPv6.