TP - application client/serveur et pair à pair

I) Mise en place

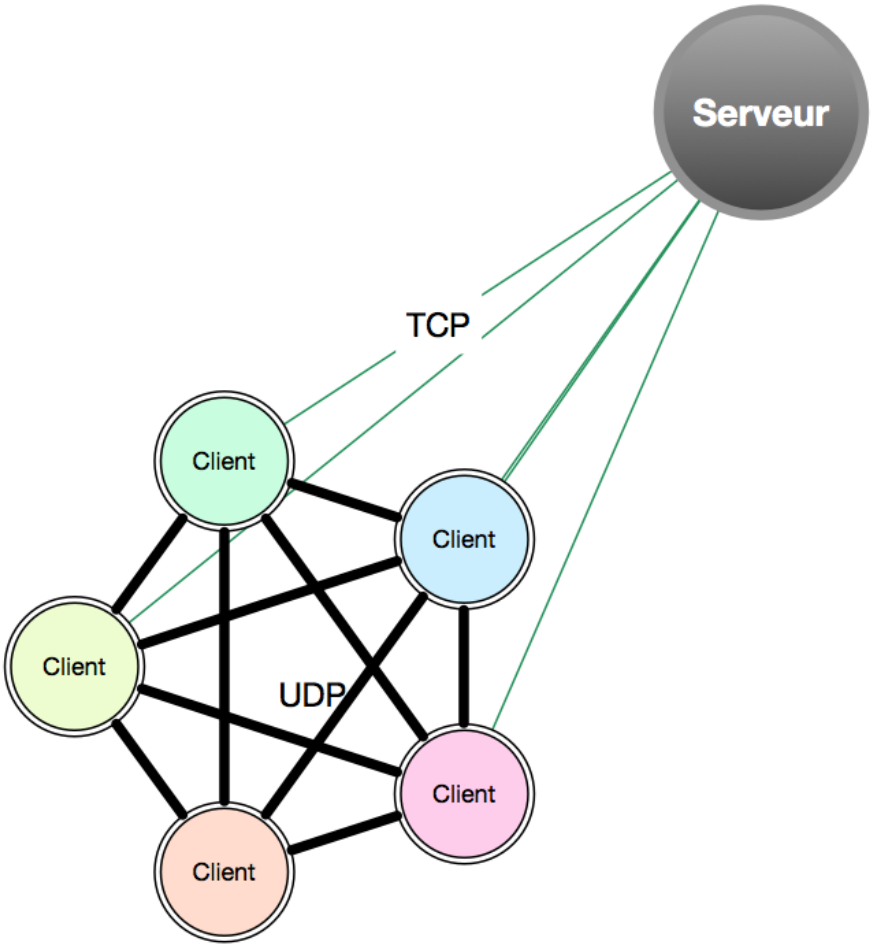
J'ai donné à chaque client une chaîne de caractères dédiée :

"Le message secret venant de" suivi de son adresse UDP

exemple : "Le message secret venant de 172.28.3.135:9904"

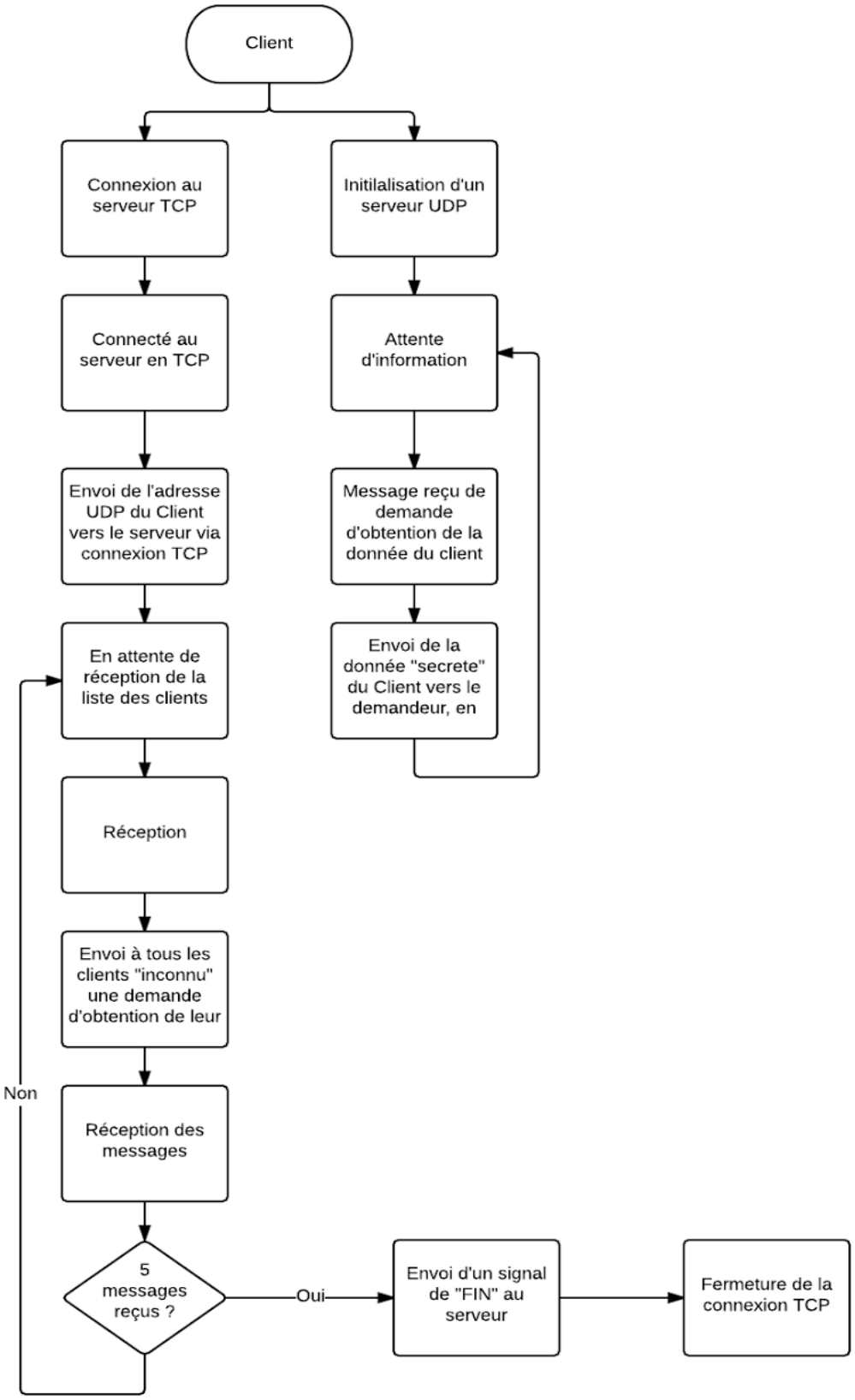
On sait que "chaque client veut obtenir 5 données", donc il nous au faudra au minimum 6 clients en même temps pour que chacun ait 5 données (je considère que la donnée personnel n'est pas à "obtenir"). Les connexions entre les clients se feront en **UDP**.

Pour obtenir la liste des autres clients, chacun d'entre eux communique avec un même serveur - celui-ci stocke donc la liste de tous les clients disponible, et gère la connexion/déconnexion de chacun d'entre eux. Les connexions entre clients et serveurs se feront en **TCP**.

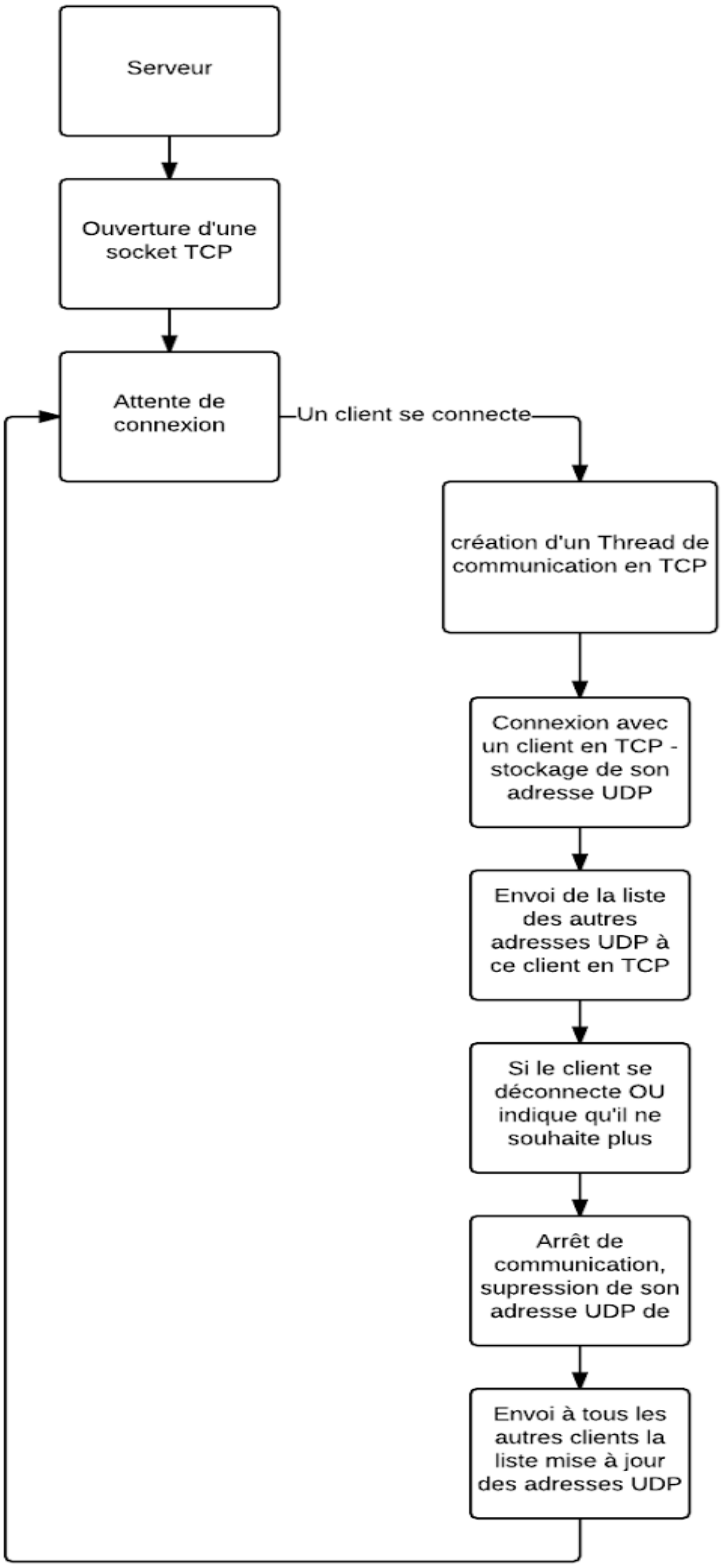


architecture des communications: en vert les connexions TCP en noir les connexions UDP

II) Client


Client
Connexion au serveur TCP
Initilalisation d'un serveur UDP
Connecté au serveur en TCP
Attente d'information
Envoi de l'adresse UDP du Client vers le serveur via connexion TCP
Message reçu de demande d'obtention de la donnée du client
En attente de réception de la liste des clients
Envoi de la donnée "secrete" du Client vers le demandeur, en
Réception
Envoi à tous les clients "inconnu" une demande d'obtention de leur
Non
Réception des messages
5 messages reçus ?
Oui
Envoi d'un signal de "FIN" au serveur
Fermeture de la connexion TCP

III) Serveur


Serveur
Ouverture d'une socket TCP
Attente de connexion
Un client se connecte
création d'un Thread de communication en TCP
Connexion avec un client en TCP - stockage de son adresse UDP
Envoi de la liste des autres adresses UDP à ce client en TCP
Si le client se déconnecte OU indique qu'il ne souhaite plus
Arrêt de communication, supression de son adresse UDP de
Envoi à tous les autres clients la liste mise à jour des adresses UDP

IV) Compilation & Exécution

a) Compilation

Les fichiers doivent être compilé avec l'option **-encoding UTF-8,** avec le JDK de version 1.8.0_05.

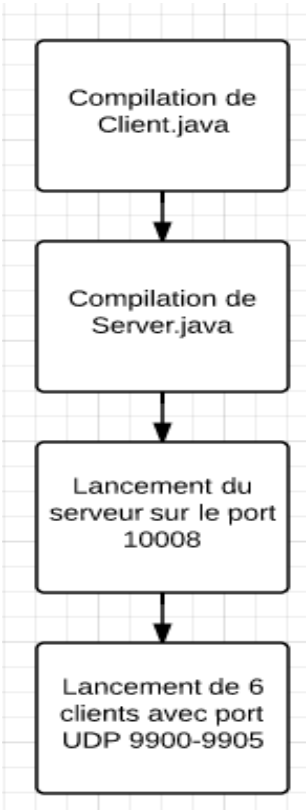
b) Exécution

Les fichiers possèdent chacun une option "**-h**" permettant de connaître la syntaxe d'éxécution.

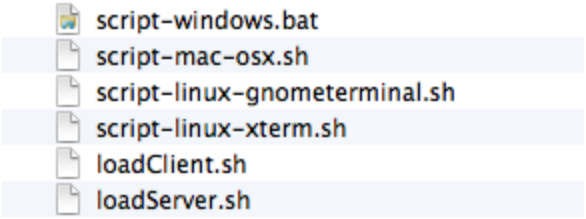
```
➜ TPSOCKETFINAL  java Server -h
Syntax: java Server PORT
➜ TPSOCKETFINAL  java Client -h
Syntax : java Client IPTCPSERVER:PORT UDPCLIENTPORT
Exit
```

Pour le serveur, on doit préciser le port que l'on souhaite.
Pour le client on doit préciser le port et l'ip du serveur, et le port UDP souhaité d'utilisation.

c) Tests



Pour tester notre application pair à pair, des scripts bash et batch ont été créé, pour chaque système d'exploitation.



Les fichiers *loadClient.sh* et *loadServer.sh* ont du être créé du fait des limitations d'OSX pour lancer java dans une autre console.

Pour pouvoir éxécuter un fichier *.sh, ne pas oublier chmod +x *.sh.

V) Programmation

a) Serveur

La classe hérite de Threads.
un “main”, créé un serveur TCP sur le port donné, et à chaque connexion créé un nouvel objet “Serveur” (Thread de communication TCP)
Le stockage de la table des clients et des adresses UDP (utilisation d’une HashMap) doit donc être “**static**”.

b) Client

Le “main” créé un thread de serveur UDP pendant qu’une connexion TCP vers le serveur se fait.
L’obtention de la “mise à jour” de la liste des adresses UDP obtenue via le serveur ainsi que le serveur UDP propre au client sont donc en parallèle.

VI) Exemples d’Output

Client

```
T-Client>Tentative de connexion ? l'h?te: 127.0.0.1 port: 10008
U-Serveur>Initilisation du serveur udp du client : 172.28.3.135:9905
U-Serveur>Message de /172.28.3.135:61531@[Demande] Le serveur 9904 veut votre message.
U-Serveur>Envoi du message de notre serveur ? /172.28.3.135:61531
U-Serveur>Message de /172.28.3.135:61532@[Demande] Le serveur 9901 veut votre message.
U-Serveur>Envoi du message de notre serveur ? /172.28.3.135:61532
U-Serveur>Message de /172.28.3.135:61533@[Demande] Le serveur 9900 veut votre message.
U-Serveur>Envoi du message de notre serveur ? /172.28.3.135:61533
U-Serveur>Message de /172.28.3.135:61534@[Demande] Le serveur 9902 veut votre message.
U-Serveur>Envoi du message de notre serveur ? /172.28.3.135:61534
U-Serveur>Message de /172.28.3.135:61535@[Demande] Le serveur 9903 veut votre message.
U-Serveur>Envoi du message de notre serveur ? /172.28.3.135:61535
T-Client>obtention de la liste des clients :
 [172.28.3.135:9904, 172.28.3.135:9901, 172.28.3.135:9900, 172.28.3.135:9905, 172.28.3.135:9902, 172.28.3.135:9903]
U-Client>Envoi de 45 bytes vers /172.28.3.135:9904
U-Client>Attente du packet de retour....
U-Client>Le serveur /172.28.3.135:9904 a envoy? :
Le message secret venant de 172.28.3.135:9904
U-Client>Envoi de 45 bytes vers /172.28.3.135:9901
U-Client>Attente du packet de retour....
U-Client>Le serveur /172.28.3.135:9901 a envoy? :
Le message secret venant de 172.28.3.135:9901
U-Client>Envoi de 45 bytes vers /172.28.3.135:9900
U-Client>Attente du packet de retour....
U-Client>Le serveur /172.28.3.135:9900 a envoy? :
Le message secret venant de 172.28.3.135:9900
U-Client>Envoi de 45 bytes vers /172.28.3.135:9902
U-Client>Attente du packet de retour....
U-Client>Le serveur /172.28.3.135:9902 a envoy? :
Le message secret venant de 172.28.3.135:9902
U-Client>Envoi de 45 bytes vers /172.28.3.135:9903
U-Client>Attente du packet de retour....
U-Client>Le serveur /172.28.3.135:9903 a envoy? :
Le message secret venant de 172.28.3.135:9903
T-Client> Envoi du signal de fin de traitement au serveur...
Client>Liste des messages re?us :
- Le message secret venant de 172.28.3.135:9904
- Le message secret venant de 172.28.3.135:9901
- Le message secret venant de 172.28.3.135:9900
- Le message secret venant de 172.28.3.135:9902
- Le message secret venant de 172.28.3.135:9903
```

La lettre “U” correspond à UDP, et la lettre T à TCP.

Serveur

```
Cr?ation de la connexion Socket:10008
En attente de connexion....
Le serveur a bien re?u l'adresse UDP 172.28.3.135:9900
{Socket[addr=/127.0.0.1,port=54313,localport=10008]=172.28.3.135:9900}
Ikezukuri d'un nouveau thread de communication
En attente de connexion....
Le serveur a bien re?u l'adresse UDP 172.28.3.135:9901
{Socket[addr=/127.0.0.1,port=54314,localport=10008]=172.28.3.135:9901, Socket[addr=/127.0.0.1,port=54313,
En attente de connexion....
Ikezukuri d'un nouveau thread de communication
Le serveur a bien re?u l'adresse UDP 172.28.3.135:9902
{Socket[addr=/127.0.0.1,port=54314,localport=10008]=172.28.3.135:9901, Socket[addr=/127.0.0.1,port=54313,
En attente de connexion....
Ikezukuri d'un nouveau thread de communication
Le serveur a bien re?u l'adresse UDP 172.28.3.135:9903
{Socket[addr=/127.0.0.1,port=54314,localport=10008]=172.28.3.135:9901, Socket[addr=/127.0.0.1,port=54313,
35:9903}
En attente de connexion....
Ikezukuri d'un nouveau thread de communication
Le serveur a bien re?u l'adresse UDP 172.28.3.135:9904
{Socket[addr=/127.0.0.1,port=54317,localport=10008]=172.28.3.135:9904, Socket[addr=/127.0.0.1,port=54314,
35:9902, Socket[addr=/127.0.0.1,port=54316,localport=10008]=172.28.3.135:9903}
En attente de connexion....
Ikezukuri d'un nouveau thread de communication
Le serveur a bien re?u l'adresse UDP 172.28.3.135:9905
{Socket[addr=/127.0.0.1,port=54317,localport=10008]=172.28.3.135:9904, Socket[addr=/127.0.0.1,port=54314,
35:9905, Socket[addr=/127.0.0.1,port=54315,localport=10008]=172.28.3.135:9902, Socket[addr=/127.0.0.1,por
En attente de connexion....
Ikezukuri d'un nouveau thread de communication
Le client /127.0.0.1:54317 a bien re?u ses 5 messages, et pr?viens le serveur
Le client /127.0.0.1:54314 a bien re?u ses 5 messages, et pr?viens le serveur
Le client /127.0.0.1:54313 a bien re?u ses 5 messages, et pr?viens le serveur
un client est parti ... sniiiiif
Le client /127.0.0.1:54316 a bien re?u ses 5 messages, et pr?viens le serveur
un client est parti ... sniiiiif
un client est parti ... sniiiiif
Le client /127.0.0.1:54315 a bien re?u ses 5 messages, et pr?viens le serveur
un client est parti ... sniiiiif
un client est parti ... sniiiiif
Le client /127.0.0.1:54318 a bien re?u ses 5 messages, et pr?viens le serveur
un client est parti ... sniiiiif
```

La liste des clients est affiché lorsqu'il y en a un nouveau