



Projet Messagerie

{ Fondamentaux des Applications Réparties }



- Livrable V1 -

Par HAKEMI Ayoub & BELKHIR Salim & AMZERT Ines



Livrable, itération 1

L'objectif de cette itération était de mettre en oeuvre la programmation réseau en utilisant l'interface socket pour le mode connecté TCP afin de créer une application (messagerie) avec un serveur relaie des messages textuels entre deux clients

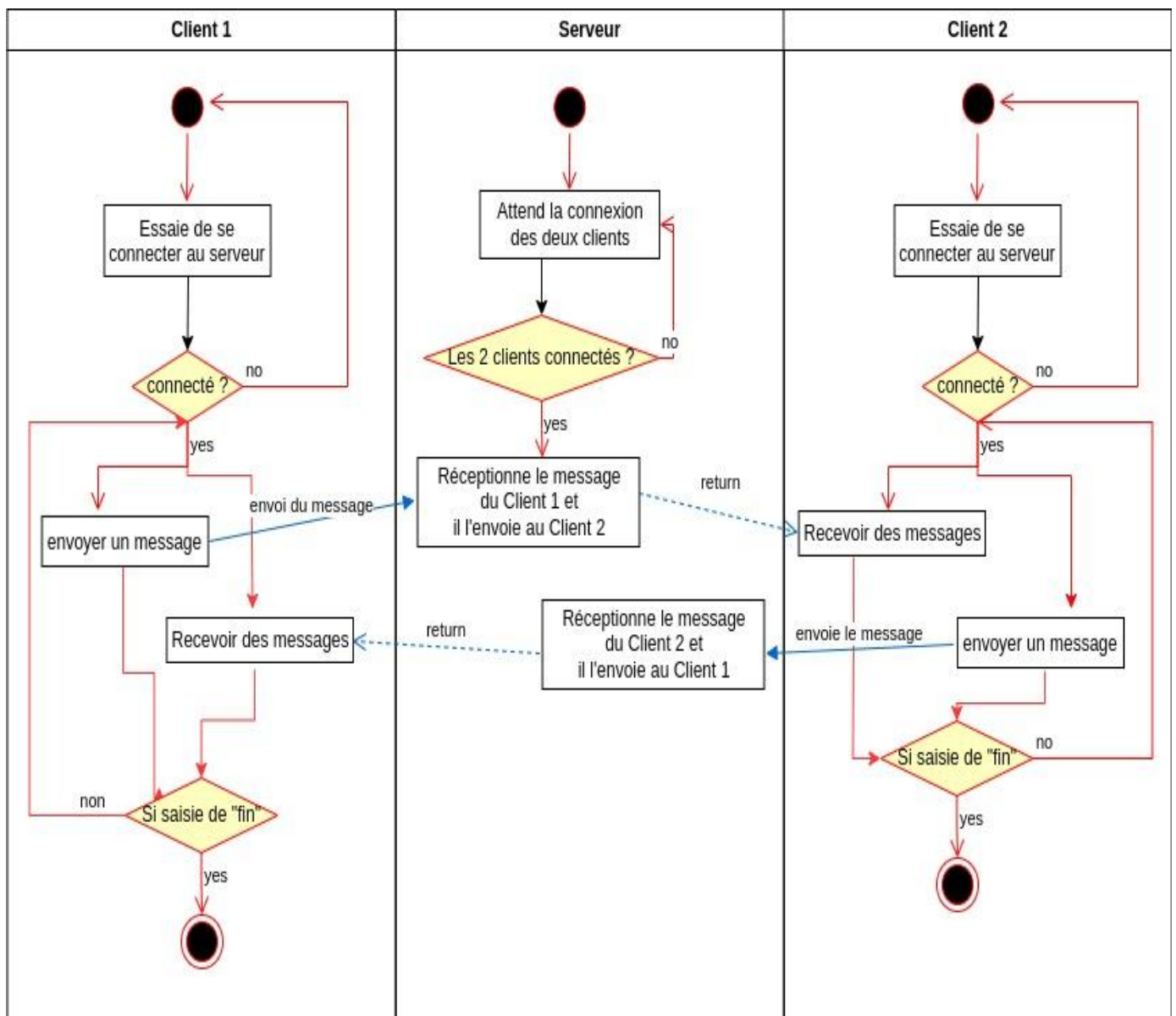
1 programme serveur et 1 programme client, lancé deux fois (deux processus distincts). Le client et le serveur sont multi-threadés.

le code source est disponible sur le répertoire git

→ <https://github.com/Salim-belkhir/Projet-FAR.git>

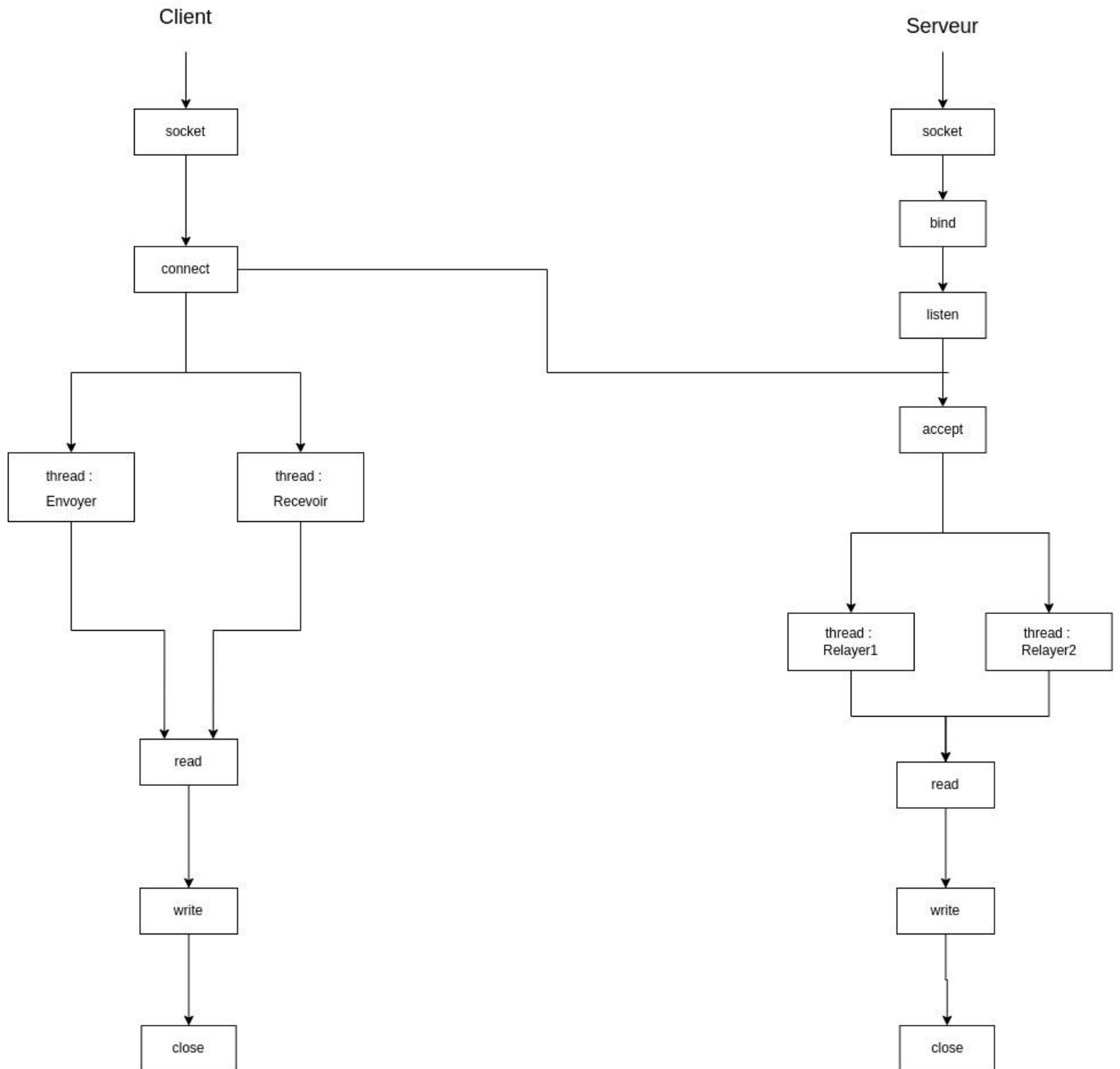


Protocole de communication entre les clients et les serveurs





L'architecture de l'application





Difficultés rencontrées

Dans un premier temps, on a dû rattraper notre retard par rapport aux cours du à une grande charge de travail (projet Web, ...), Nous avons aussi dû revoir les bases comme les threads, les processus, etc...

La réalisation de la déconnexion des clients et l'arrêt de leur thread manipulant le relais nous pose toujours un problème que l'on va résoudre lors du prochain sprint. Pour l'instant, notre application permet seulement à deux clients de communiquer correctement. Il faut qu'ils se déconnectent pour permettre à d'autres clients de communiquer. Par contre, tous les clients peuvent se connecter en même temps mais seulement les deux premiers pourront communiquer. Donc la gestion du serveur n'était pas si évidente pour nous et on compte l'améliorer.



Répartition des tâches

Remise à niveau pour tout le groupe. On a essayé chacun de notre côté de faire la base seul, pour être obligés de travailler.

Après, nous avons décidé de travailler en groupe pour faire une version plus optimale, pour pouvoir s'aider mutuellement et mettre en commun les idées et améliorer l'application.

On s'est concertés tous ensemble pour rédiger le rapport. Chacun a un peu touché à tout.



Commandes afin de compiler et exécuter le code

Clients À tour de rôle

Compiler : `gcc -o client(1ou2) client(1ou2).c -lpthread`

Exécuter → `./client1 ton-adresse-ip le-port-du-serveur`

Exécuter → `./client2 ton-adresse-ip le-port-du-serveur`

Client SANS tour de rôle

Compiler : `gcc -o client4 client4.c -lpthread`

Exécuter → `./client4 ton-adresse-ip le-port-du-serveur`

Serveur

Compiler : `gcc -o server serve.c -lpthread`

Exécuter → `./serveur ton-adresse-ip le-port-du-serveur`

on peut créer un serveur pour faire les testes avec
nc -l -p 5000
5000 : c'est le numéro du port