# Rapport de projet réseau

## Université de Strasbourg

Professeur:

Julien MONTAVONT

### Table des matières

Introduction	
Machine à états finale	4
Mode binaire	4
Transfert de fichier	5
Gestion avancée des utilisateurs	5
Conclusion	6

#### Introduction

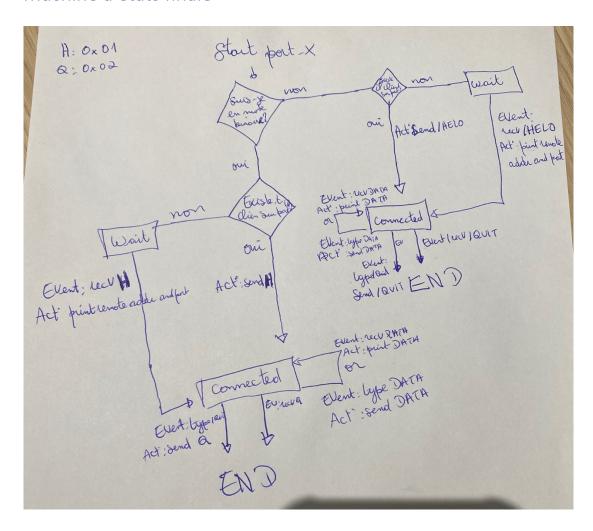
Pour le cours de réseau du semestre 5 nous avions eu comme projet d'implémenter un chat entre deux ou N utilisateurs et compatibles IPV4/IPV6. Dans sa version basique le programme ne se contente de faire que du transfert brut de texte.

J'ai pu implémenter cette version en me basant sur mes acquis de l'exercice 12 des tp's effectués en cours.

Pour aller plus loin, il nous a été demandé d'ajouter des fonctionnalités supplémentaires à ce chat afin d'en améliorer la performance. Notamment un protocole en mode binaire pour réduire la taille des informations de contrôle. En effet , lorsque nous. Sommes dans la version basique, les commandes ainsi que les données sont envoyées en format ASCII ce qui alourdit la communication. En mettant en place ce protocle on peut s'affranchir de l'envoi des commandes en format ASCII. Protocole que j'ai réussi à mettre en place.

D'autres fonctionnalités nous ont aussi été demandées comme le transfert de fichier et l'utilisation de N utilisateurs en simultanée du chat.

#### Machine à états finale



#### Mode binaire

Afin de mettre en place le protocole en mode binaire pour gérer les commandes /HELO et /QUIT, j'ai pris parti d'utiliser les directives préprocesseurs **H** et **Q** codé sur un octet représentant respectivement les commandes /HELO et /QUIT. H définit le binaire 0x01 et **Q** définit 0x02.

J'ai aussi défini une strucure **struct messageB**, cette structure contient un seul champ **msgB** de taille uint8 t.

Lorsque le client veut envoyer la commande /HELO, il l'écrit dans le buffer. Je récupère donc la commande dans le buffer et au lieu d'envoyer toute la commande, je fais usage de mes directives préprocesseurs ainsi que de ma structure message afin de convertir cette commande en binaire et de l'envoyer. De même lorsque le serveur reçoit les commandes en binaire H ou Q il sait exactement comment réagir. Le client n'a pas la visibilité sur l'envoi. Pour lui il ne fait toujours qu'écrire les commandes de base /HELO et /QUIT.

J'ai choisi de le coder sur uint8\_t car c'est la plus petite taille de donnée (non booléenne) que l'on peut allouer. Il s'écrit sur exactement 1 octet.

#### Transfert de fichier

#### Non implémenté

Ce protocole permettrait aux utilisateurs d'envoyer et de recevoir des fichiers en mode texte et/ou binaire. En effet, dans la version basique les utilisateurs ne peuvent s'échanger que du texte. Cette implémentation vient renforcer la communication.

#### Gestion avancée des utilisateurs

#### Non implémenté

Ce protocole permettrait à plusieurs utilisateurs de communiquer de manière simultanée en sachant à chaque fois qui a envoyé le message. En effet, dans sa forme basique le programme ne permettait que la communication entre seulement deux utilisateurs. J'avais pensé à rajouter à cette partie un message lorsqu'un utilisateur laisse la conversation mais aussi lorsqu'un utilisateur rentre dans la conversation. Cela permettrait aux autres de savoir combien ils sont et qui rentre/sort du chat.

#### Conclusion

Ce projet a été l'occasion pour moi de revenir sur les notions vues en réseau ce semestre et de mettre en avant mes capacités algorithmiques. J'ai pu effectuer pour la première partie du projet (en mode basique) les tests nécessaires donnés à l'exercice 12. J'ai pu aussi implémenter le protocole en mode binaire. Cependant, il m'a été beaucoup plus difficile d'implémenter les deux autres protocoles. C'est ce point faible que. Je. Trouve à mon projet. Cependant même si je n'ai pas pu l'implémenter pour la deadline, je continuerai pendant les vacances de travailler dessus afin d'avoir un projet complètement abouti pour la rentrée. Ce projet fut un vrai challenge et j'ai aimé le faire même si. J'y intégrerai aussi à l'avenir d'autres fonctionnalités auxquelles je réfléchirai aussi afin de mieux préparer les UE de la S6 notamment le projet intégrateur à venir.