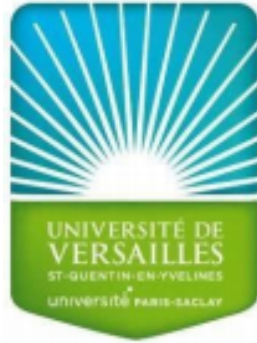


Université de Versailles Saint-Quentin-en-Yvelines  
UFR des sciences  
Département Informatique



# **SIMULATION DU PROTOCOLE TCP**

## Rapport Final du projet

ALHADJ ADAM Mamadou  
N°Etudiant : 21920273  
SY Cherif Amanatoulha  
N°Etudiant :21921092

28 mai 2021

# Table des matières

|   |          |
|---|----------|
| <b>Préambule</b>                                | <b>3</b> |
| 0.1 Définition précise du sujet . . . . .       | 3        |
| 0.2 Architecture de l'application . . . . .     | 3        |
| 0.3 Prérequis . . . . .                         | 3        |
| 0.4 Fonctionnement des modules . . . . .        | 3        |
| 0.4.1 Serveur . . . . .                         | 3        |
| 0.4.2 Client . . . . .                          | 3        |
| 0.4.3 TimeOut . . . . .                         | 3        |
| 0.4.4 Buffer . . . . .                          | 3        |
| 0.5 Fonctionnement de l'application . . . . .   | 3        |
| 0.6 Choix du langage de programmation . . . . . | 3        |
| 0.7 Difficultés rencontrés . . . . .            | 3        |
| 0.8 Conclusion . . . . .                        | 3        |
| 0.9 Conclusion Technique . . . . .              | 3        |
| 0.10 Conclusion Organisationelle . . . . .      | 3        |
| <b>1 Définition precise du Sujet</b>            | <b>4</b> |
| 1.1 Protocole TCP . . . . .                     | 4        |
| 1.2 Architecture de l'application . . . . .     | 4        |
| 1.2.1 Client . . . . .                          | 4        |
| 1.2.2 Serveur . . . . .                         | 4        |
| 1.3 Prérequis . . . . .                         | 4        |
| <b>2 Fonctionnement des modules</b>             | <b>5</b> |
| 2.1 Modules Serveur . . . . .                   | 5        |
| 2.2 Module Client . . . . .                     | 5        |
| 2.3 TimeOut . . . . .                           | 5        |
| 2.4 Buffer . . . . .                            | 5        |
| <b>3 Fonctionnement de l'application</b>        | <b>6</b> |
| 3.1 Choix du langage de programmation . . . . . | 6        |
| 3.2 Difficultés rencontrés . . . . .            | 7        |
| 3.3 Difficultés Techniques . . . . .            | 7        |
| 3.4 Difficultés organisationnelles . . . . .    | 7        |
| <b>4 Conclusion</b>                             | <b>8</b> |
| 4.1 Conclusion technique . . . . .              | 8        |
| 4.2 Conclusion organisational . . . . .         | 8        |

## Préambule

- 0.1 Définition précise du sujet
- 0.2 Architecture de l'application
- 0.3 Prérequis
- 0.4 Fonctionnement des modules
  - 0.4.1 Serveur
  - 0.4.2 Client
  - 0.4.3 TimeOut
  - 0.4.4 Buffer
- 0.5 Fonctionnement de l'application
- 0.6 Choix du langage de programmation
- 0.7 Difficultés rencontrés
- 0.8 Conclusion
- 0.9 Conclusion Technique
- 0.10 Conclusion Organisationelle

# 1 Définition précise du Sujet

## 1.1 Protocole TCP

Le protocole TCP/IP est mis en place suite à une réflexion de chercheurs américains à l'issu d'un problème posé par le département de la défense (DoD), cette dernière disposait de plusieurs bases militaires sur le territoire, et chacune de ces bases disposait de sa propre logistique informatique.

Cependant ces différentes bases peuvent juste partager leurs informations locales et ne peuvent communiquer entre elles. Pour les relier, il a fallu trouver un système permettant de le faire. De ce fait le Protocole TCP/IP est né.

TCP/IP pour Transmission Contrôle Protocol / Internet Protocol, est une suite de protocoles constituant l'ensemble des règles permettant la transmission de données sur un réseau. TCP prend en charge l'ouverture et le contrôle de la liaison entre des ordinateurs et le protocole d'adressage IP assure quant à lui le routage des données transmises. Dans les spécifications de notre projet ,nous avons pu implémenter notre Simulateur TCP/IP en respectant les besoins attendus par l'enoncé du projet.

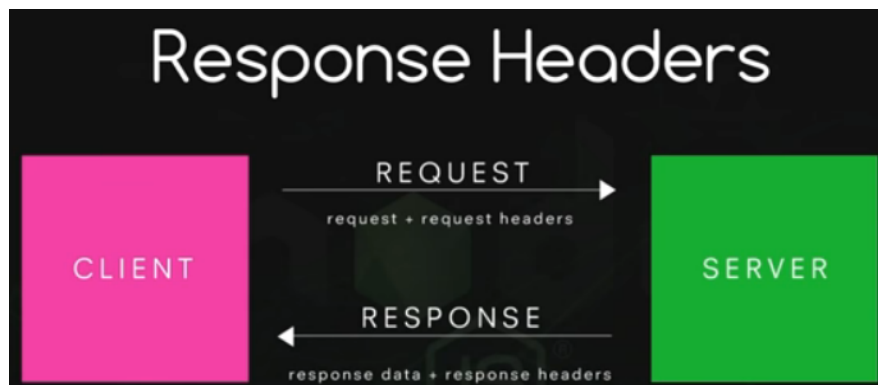
Notre simulateur TCP/IP permettra de façon simplifiée à simuler un échange des paquets de bout en bout entre un client et un serveur.

Le client ouvre la connexion, le serveur le renvoie le nombre demandé de paquets par le client, ce dernier fermera la connexion dès réception des paquet.

Cette implementation à suivi plusieurs étapes que nous tenterons d'expliquer dans les moindres détails dans la suite de ce rapport.

## 1.2 Architecture de l'application

l'application décrit une architecture assez simple, composée essentiellement de deux grandes parties notamment un client et un serveur :



### 1.2.1 Client

Il s'agit d'une partie integrante ou le compatriement de l'application chargé d'envoyer des requetes(demandes) au serveur. Il contient également toutes les instructions relatives à la formulation d'une requete au serveur auquel il est connecté.

### 1.2.2 Serveur

Compatriement essentiel dans la communication,le serveur désigne le rôle joué par un appareil matériel destiné à offrir des services à des clients en réseau Internet ou intranet.

## 1.3 Prérequis

Afin d'assurer le bon fonctionnement de l'application, l'utilisateur doit avoir vu certaines notions fondamentales des réseaux informatiques, pour pouvoir utiliser au mieux les fonctionnalités proposés par le simulateur. L'utilisation de cette simulation cependant ne necessite pas forcément une très grande connaissance au préables du protocles TCP/IP.Cependant,Il est nécessaire de comprendre le fonctionnement de la communication d'une architecture Client/Server ; la création, la simulation et la visualisation d'un protocole TCP/IP.

De plus pour des raisons pédagogiques et de support à l'enseignement et à la compréhension des réseaux qui utilisent le protocole TCP/IP, il est important d'avoir une compréhension de la programmation.

## 2 Fonctionnement des modules

### 2.1 Modules Serveur

le module Serveur est composé de plusieurs sous modules chaque un doté d'un rôle spécifique et tous concourant à apporter des services aux clients. Dans le serveur nous retrouvons :

- La classe Serveur : elle implemente le constructeur qui spécifie le port d'écoute de la communication
- Timeout : elle implemente le délai d'attente de donnée
- Ecrire\_Client : méthode d'envoi des données aux clients
- Lire\_client : lire les données envoyées par le client
- la connexion : elle implemente la connexion avec le client
- fermeture de connexion : elle implemente la fermeture de connexion entre le serveur et le client

### 2.2 Module Client

le module client abrite plusieurs sous modules tous concourant à établir l'ensemble des règles formulant des requêtes au serveur. Dans le module client nous pouvons noter les sous modules suivants :

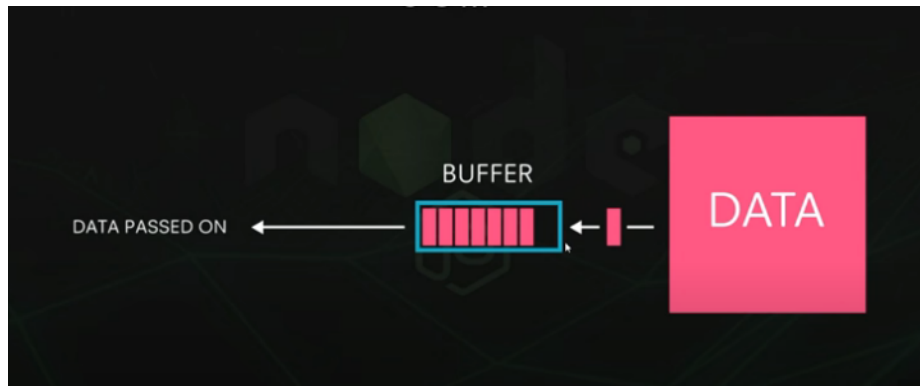
- la classe client : elle implemente le constructeur qui indique le serveur auquel le client est connecté et spécifie le port de communication utilisé.
- ecrire\_serveur : elle implemente la méthode qui permet d'envoyer des données au serveur
- lire\_serveur : elle implemente la méthode qui permet de recevoir les données envoyées par le serveur
- connexion : elle établit la connexion avec le serveur
- fermer\_connexion : ferme la communication avec le serveur

### 2.3 Timeout

Le time out correspond au dépassement d'un délai accordé pour l'exécution d'une certaine action. Dans notre projet pour effectuer la connexion, l'application dispose d'un certain délai fixé à priori de 30 secondes à expiration temps duquel expire le time out (temps dépassé)...les raisons peuvent être multiples : pas de connexion réseau, pas de serveur cible disponible, ressources non disponibles pour l'application, pas de client connecté etc..... Dans l'implémentation de notre application le timeout est déclenché dès la mise en fonction du serveur, celui-ci reste actif pour un temps (30 secondes) à priori défini et reste démarré sur le port de connexion (port 555) et attend l'établissement de connexion avec un client (telnet natif etc...) cette durée d'attente est prédéfinie (timeout) dès qu'elle est atteinte la connexion échoue et il faudra rétablir une nouvelle connexion.

### 2.4 Buffer

La mémoire tampon (buffer) sert à stocker temporairement des données dans la mémoire vive ou dans le disque dur de l'ordinateur. C'est en quelque sorte la "salle d'attente" des données et de toutes les informations qui transitent au sein de l'ordinateur. Sans la mémoire tampon et sans les tampons, le transfert de données fonctionnerait beaucoup moins efficacement et les temps d'attente seraient très longs. Le concept de la mémoire tampon a été mis au point pour éviter l'encombrement des données d'un port entrant à un port de transfert sortant. Dans la simulation du protocole TCP/IP le buffer est très important car permettant au serveur d'effectuer l'envoi des paquets de données sur un ou plusieurs envois selon la taille de sa capacité de réception rcvwindows.



### 3 Fonctionnement de l'application

le fonctionnement de l'application se résume sur plusieurs étapes :

- Etape 1 : Ouverture de connexion - Le Client envoie un paquet de synchronisation (SYN) au serveur et déclenche un temporisateur (TimeOut). - Le serveur le renvoie un paquet de synchronisation plus un acquittement (SYN + ACK)
- Etape 2 : Transfert de données - Le client demande au serveur de lui envoyer un nombre variable de paquets (N), en précisant sa capacité de réception (taille de sa file d'attente rcvwindow). - Le serveur compare N et rcvwindow afin de faire un ou plusieurs envois - Le client acquitte les paquets soit positivement ou négativement. En cas d'un acquittement négatif, le serveur retransmettra le ou les paquets erroné(s).
- Etape 3 : Fermeture de connexion - Le client demande au serveur de fermer la connexion dans un paquet (FIN) - Le serveur acquitte le paquet (FIN) avec un paquet (en cours de fermeture) - Le client acquitte le paquet (en cours de fermeture) et déclenche un temporisateur de 30 seconde puis ferme la connexion.

#### 3.1 Choix du langage de programmation

Réaliser un projet d'une telle envergure nécessite des choix techniques pertinents et adaptés aux besoins du projet. C'est pourquoi, face aux exigences techniques liés à la mise en place d'un modèle client/Server et les spécificités techniques dont ils font recourt, nous avons décidé objectivement d'implémenter en JAVA.

Java est un langage de programmation orienté objet. Il est utilisé dans diverses plates-formes informatiques. On le voit presque partout aujourd'hui, des appareils intégrés aux téléphones mobiles, en passant par les serveurs d'entreprise et les supercalculateurs. Aux yeux de nombreux experts en informatique, Java est une propriété essentielle de Sun Microsystems. Scrupuleusement, JAVA a apporté de nombreux avantages aux développeurs de logiciels surtout en ce qui concerne la programmation réseau.

voici quelques points que nous pouvons relaté sur le choix porté sur le langage de programmation JAVA

- C'est simple.  
Ce programme en particulier est spécialement conçu pour être très convivial. Comparé à d'autres langages de programmation, Java est plus facile à écrire, compiler, déboguer et apprendre. En effet, il utilise la gestion automatique de la mémoire et la récupération de place. Les nouveaux apprenants peuvent l'appliquer facilement dans un court laps de temps.
- Il est indépendant de la plate-forme  
La meilleure chose à propos de Java est sa capacité à passer facilement d'un système informatique à un autre sans créer de problème. "Écrire une fois, courir n'importe où" décrit le mieux Java. Il peut fonctionner indépendamment aux niveaux source et binaire. Le même programme peut être exécuté sans problème sur différents systèmes. Cette fonctionnalité spéciale est extrêmement essentielle pour les logiciels World Wide Web, qui nécessitent beaucoup de flexibilité.
- Il est sécurisé  
Si la sécurité est votre principale préoccupation, Java sera votre meilleur choix. Il place la sécurité au premier rang de ses priorités. Son langage, son compilateur, son interpréteur et son environnement d'exécution sont personnalisés dans un souci de sécurité. Sa plate-forme permet aux utilisateurs de télécharger du code inconnu sur un réseau et de l'exécuter dans un environnement sécurisé sans nuire aux systèmes. Le système hôte ne sera pas du tout infecté par le virus. Cette capacité à elle seule a rendu la plate-forme Java unique.

- il est fiable

Honnêtement, il est en effet difficile de rechercher des langages de programmation vraiment robustes. Cependant, Java a déployé beaucoup d'efforts pour vérifier au préalable les erreurs éventuelles. Les compilateurs Java sont capables de détecter de nombreux problèmes lors de la phase initiale d'exécution. Par conséquent, il est considéré comme la programmation la plus fiable par les développeurs.

- Il peut effectuer plusieurs tâches.

Java a la capacité d'effectuer plusieurs tâches simultanément. La programmation multithread s'y est bien intégrée. Cela fonctionne mieux dans la programmation visuelle et réseau. Parallèlement, ses normes stables aident les développeurs à créer des applications à plusieurs niveaux avec une approche par composants.

- C'est dynamique

Le code Java est organisé en unités modulaires orientées objet. Ces unités sont appelées classes. Ils sont stockés dans des fichiers distincts et chargés dans l'interpréteur Java uniquement lorsque cela est nécessaire. Dans cette situation, un programme peut s'étendre de manière dynamique en chargeant les classes dont il a besoin pour étendre ses fonctionnalités.

- C'est économique

En terme d'investissement dans la programmation Java, rien à dépenser ? C'est une source ouverte. Par conséquent, nous n'avons pas besoin de lutter contre des frais de licence élevés chaque année.

## 3.2 Difficultés rencontrés

Réaliser en autonomie un projet guidé, est loin d'être une tâche simple car cela demande une excellente autonomie et une volonté sans failles, d'aller chercher des techniques et outils largement au delà de l'information contenue dans les Cours. C'est pourquoi les difficultés sont inévitables et cela nous reviens de coutume d'en faire face et de trouver des solutions efficaces et adaptés à nos difficultés. Ainsi, au cours de la réalisation du projet, nous avons rencontrés plusieurs difficultés que nous pouvons classer en deux catégories :

- les difficultés techniques
- les difficultés organisationnelles

## 3.3 Difficultés Techniques

Les difficultés d'ordre techniques s'étalent sur plusieurs formes partant de la conception de l'application jusqu'à son déploiement passant par son développement et l'ensembles des outils techniques utilisés. la conception de l'application avait confronté quelques difficultés liés au transfert de données entre le client et le serveur. ces difficultés sont également noté sur l'adaptabilité des sockets.

Nous avons également rencontré des difficultés liées au code notamment l'utilisation et l'intégration des temporisateurs lors de la fermeture de connexion entre le client et le serveur .

Nous pouvons également les difficultés relatives à la transmission des variables entières dans un flux de données de caractère qu'on envoie depuis le client vers le serveur et de pouvoir effectuer le calcul pour le transfert des données.

## 3.4 Difficultés organisationnelles

Pour mener à bien notre travail et de pouvoir impliquer tous les membres du groupe dans le projet, l'organisation et la bonne communication interne étaient les véritables paris qu'il fallait nécessairement réussir. La réussite de celui-ci passe par d'énormes difficultés d'organisation.

En effet nous avons le problème de converger nos horaires pour travailler ensemble car n'ayant pas suivi le même parcours. la diversité de nos parcours faisait que nous n'avions pas les mêmes horaires de travail (Heure de cours, ou TD), il nous était donc très difficile d'organiser des séances de travail qui réunissaient tous les membres du groupe en un même créneau. Ainsi, pour pallier à ces difficultés nous avons utilisé des outils comme Discord pour la communication et le HACKNPLAN pour les planifications et les assignations de tâches aux participants, mais également de suivre l'avancement du projet en même temps.

Nous ne pouvons pas terminer les problèmes d'organisation sans souligner le contexte sanitaire (COVID 19) dans lequel nous vivons depuis plus d'un an déjà. Cette période très compliquée a rendu les choses plus compliquées car nous avons des problèmes pour se regrouper en présentiel pour travailler ensemble et ceci était un véritable problème pour savoir comment nous avançons et se sentions impliquer dans le projet.

## 4 Conclusion

### 4.1 Conclusion technique

Le produit final obtenu après les phases du projet, est très proche du résultat attendu en terme de fonctionnalités. Toutes les fonctionnalités décrites dans les section précédentes sont implémentés et fonctionnelles. Cependant, quelques "bugs" peuvent s'être glissés dans le module de tranferts de donées, ces "bugs" seront corrigés entre la remise du projet et la soutenance. Leur résolution sera indiqué lors de la démo. D'autre part, pendant l'étape de programmation des potentielles idées d'amélioration ont été suggérés et retenues par les membres du groupe et leur implémentation se fera entre la remise et la soutenance en fonction du temps alloué à chaque tâche.

### 4.2 Conclusion organisationnel

La réalisation de ce projet nous a permis dans un premier temps de gérer le travail en équipe. Cela n'a pas toujours été facile, puisqu'il a fallu faire des mises au point, des réunions, comprendre ce que les autres personnes du groupe faisaient, et arriver à nous mettre d'accord sur les différentes décisions prises durant le traitement des différents aspects du projet. Ce genre de projet en équipe permet d'apprendre à comprendre les choix des autres, faire des critiques constructives, accepter ses erreurs et se remettre en question. D'autre part, Le projet tutoré nous a également permis de mettre à profit nos connaissances sur un sujet où est réalisé l'une des plus grandes recherches de l'histoire de l'informatique, et ça nous a permis d'apprendre les différents concepts que ce soit dans le domaine des réseaux, de la programmation, de l'algorithmique et de la gestion de projet. Un projet d'une telle envergure apporte beaucoup, par exemple dans la gestion du temps, car le respect de l'échéancier n'est pas toujours évident. Pour veiller au bon respect des échéancier sur les tâches de chacun, de la bonne entente entre les membres et de leurs communication, nous avons élu un chef de projet et ce dès la première semaine. Ce projet a donc été très instructif pour tous les membres du groupe. Cette expérience nous a appris que la place de la programmation dans un projet logiciel est bien importante qu'on le pensait au départ, et que les étapes de conception sont tout aussi importantes et déterminantes dans la réalisation d'un projet logiciel.