

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université 8Mai 1945 – Guelma
Faculté des Sciences et de la Technologie
Département d'Electronique et Télécommunications



**Mémoire de Fin d'Etude
Pour l'Obtention du Diplôme de Master Académique**

Domaine : **Sciences et Techniques**
Filière : **Télécommunications**
Spécialité : **Réseaux de Télécommunications**

**Réalisation d'une application client serveur
en Java**

Présenté par :
BENZEKRI Seyf eddine
AFRI Younes

Sous l'encadrement du :
Dr. HALASSI Abderezzaq

Année Universitaire : 2020-2021

REMERCIEMENT

Nous tenons tout d'abord à remercier *ALLAH* le tout Puissant et Miséricordieux.

Qui nous a donné la force et la patience pour accomplir ce travail.

EL HAMDOULILLAH.

Nous tenons à remercier *Docteur HALLASSI Abderezzaq* notre encadreur pour avoir bien voulu diriger notre projet de fin d'étude et surtout la confiance qu'ils nous a accordée sans oublier bien sur ses conseils et ses encouragements le long de ce projet.

Nous voudrions également remercier tous les profs qui nous ont aidés à accomplir ce modeste travail.

Nos vifs remerciements vont également aux *membres du jury* pour l'intérêt qu'ils ont porté à notre travail en acceptant de l'examiner et de l'enrichir.

Nos remerciements vont également chez tous nos enseignants du département Réseaux de télécommunication.

Nous souhaitons à tous nos *collègues de Master* une bonne continuation vers la réussite.

Merci à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Dédicace 1

Je dédie ce modeste travail en premier lieu

A mon père et ma chère mère,

*A qui je leur dois la réussite pour l'éducation
qu'ils m'ont prodiguée ;*

*Et à mes frères Imad, Fouad et Chaouki
et ma sœur Siham, que je leurs souhaite une
vie Pleine de réussites,*

*Ainsi qu'à toute ma famille et mes amis pour
leurs patiences*

Et bien sûr à mon binôme Younes et sa famille.

*Egalement à notre encadreur Dr. Halassi
Qui nos a toujours apporté son précieux
Conseils et sa confiance.*

*Et à tous les étudiants de la promotion 2021,
notamment ceux qui me connaissent.*

SAYF EDDINE

Dédicace 2

Je dédie ce modeste travail

*À mes parents qui m'ont toujours apporté
leurs précieux conseils, leur confiance et*

Surtout leur soutien moral.

*Et à mes chères sœurs et frères, que je leurs
souhaite une vie Pleine de réussites.*

Ainsi que toute ma famille

A tous mes amis sans oublier mon binôme

Saïf eddine et sa famille.

*Aussi à notre promoteur Dr. Halassi, qui nos a
toujours apporté leurs précieux conseils et*

surtout sa confiance

*Et à toute la promotion 2021 sans oublier tous
les étudiants de notre université*

YOUNES

RÉSUMÉ

Le but de ce travail était d'une part l'étude théorique sur les réseaux de télécommunication notamment les plus utilisés à savoir leurs architectures, les protocoles utilisés et comment fonctionnent les types de réseaux existents (LAN, WAN, VPN). En effet on a été amené à s'intéresser essentiellement au modèle de base des réseaux informatiques qui est le modèle client-serveur afin d'expliquer comment créer un réseau local LAN grâce à l'utilisation des sockets. D'autre part, il résume notre démarche en la réalisation d'une application en Java de gestion administrative pour les étudiants et les enseignants, qui leur offre un choix de communication intéressant, facile et rapide.

Mots clés : réseau informatique, système de télécommunication, réseau local, socket, application administrative.

ABSTRAC

The object of this work is, on the one hand, the theoretical study of telecommunications networks, in particular the most widely used telecommunications networks, i.e. their architecture, the protocols used and the way in which the types of networks (LAN, WAN, VPN) exist. In fact, we are mainly interested in the basic model of computer networks, the client-server model, to explain how to create a local area network LAN using sockets. On the other hand, it sums up our approach by creating a Java administrative management application for students and teachers, which gives them a fun, simple and fast communication option.

Keywords: computer network, telecommunications system, local network, socket, management application.

المخلص

إن الهدف من هذا العمل من جهة هو الدراسة النظرية لشبكات الاتصالات السلكية واللاسلكية، ولا سيما الأكثر استخداماً، أي بنيتها والبروتوكولات المستخدمة و كيفية عمل مختلف أنواع الشبكات (LAN، WAN، VPN). وكذلك كنا مهتمين بشكل كبير بالنموذج الأساسي لشبكات الكمبيوتر وهو نموذج الخادم والعميل وهذا لشرح كيفية إنشاء شبكة محلية LAN من خلال استخدام المقابس.
من ناحية أخرى، قمنا بإنشاء تطبيق إداري بواسطة Java للطلاب و الأساتذة، والذي يسهل لهم عملية التسجيل والتواصل فيما بينهم.

الكلمات المفتاحية : شبكة كمبيوتر، نظام اتصالات، شبكة محلية، مقبس، تطبيق إداري.

TABLE DES MATIERES

REMERCIEMENT.....	ii
TABLE DES MATIERES.....	vi
LISTES DES FIGURES ET TABLEAUX.....	ix
ABREVIATIONS ET SYMBOLES.....	xi
INTRODUCTION GENERALE.....	1
CHAPITRE I : NOTIONS FONDAMENTALES SUR LES RESEAUX INFORMATIQUES	3
I.1. Introduction.....	3
I.2. Définition de réseau informatique.....	3
I.3. Principe de fonctionnement.....	4
I.4. Architecture client/serveur.....	4
I.4.1. Protocoles réseau.....	5
I.4.2. Adresses réseau.....	6
I.5. Types de réseaux existents.....	6
I.5.1. Mode de transmission.....	6
I.5.2. Portée.....	6
I.5.2.1. Réseau Personnel (PAN).....	6
I.5.2.2. Réseaux locaux (LAN).....	7
I.5.2.3. Réseaux Métropolitain (MAN).....	7
I.5.2.4. Réseaux Etendus (WAN).....	8
I.5.2.5. Réseaux Mondial (GAN).....	9
I.5.2.6. Réseaux Privés Virtuels (VPN).....	9
I.6. Topologie des réseaux.....	10
I.6.1. Définition de topologie.....	10
I.6.2. Topologie en bus.....	11
I.6.3. Topologie en étoile.....	11
I.6.4. Topologie en anneau.....	12
I.6.5. Topologie en arbre.....	12
I.6.6. Topologie maillée.....	13
I.7. Modèle OSI (OPEN SYSTEMS INTERNCONNECTION).....	14
I.8. Modèle TCP/IP (Transmission Control Protocol/Internet Protocol).....	15
I.9. Comparaison TCP/IP - OSI.....	15
I.10. Protocole IP (Internet Protocol).....	16
I.11. Adressage IPV4.....	16
I.11. 1. Anatomie d'une adresse IP.....	17
I.11.2. Masque de réseau.....	17

I.11.3. Classes d'adresses	18
I.12. Conclusion.....	18
CHAPITRE II : Réseau local LAN et ses protocoles.....	19
II.1. Introduction.....	19
II.2. Création d'un réseau local (LAN).....	19
II.2.1. Deux types de réseau locaux.....	19
II.2.2. Matériels nécessaires	21
II.2.3. Comment envoyer des messages entre machines dans un réseau local ?	21
II.2.4. Comment envoyer des messages entre machines dans deux réseaux différents ?.....	21
II.2.5. Comment communiquent des applications sur différentes machines ?	21
II.3. Modèle client-serveur	22
II.3.1. Définition	22
II.3.2. Caractéristiques des systèmes client-serveur	23
II.3.2.1. Service.....	23
II.3.2.2. Partage de ressources	23
II.3.2.3. Protocole asymétrique.....	23
II.3.2.4. Transparence de la localisation	23
II.3.2.5. Message.....	23
II.3.2.6. Encapsulation des services.....	23
II.3.2.7. Evolution.....	23
II.3.3. Types de modèle client serveur.....	23
II.3.3.1. Client -serveur de donnée	23
II.3.3.2. Client -serveur de présentation	24
II.3.3.3. Client –serveur de traitement	24
II.4. Notion de port	24
II.5. Socket.....	25
II.5.1. Historique.....	25
II.5.2. Définition Sockets.....	25
II.5.3. Principe de fonctionnement	26
II.6. Protocoles utilisés dans sockets	26
II.6.1. Protocole de datagramme utilisateur (UDP)	27
II.6.2. Protocole TCP.....	27
II.7. Conclusion	27
CHAPITRE III : <i>Application de gestion des étudiants et des enseignants en Java</i>	28
III.1. Introduction	28
III.2. Langages Utilisés.....	28

III.2.1. Langage JAVA	28
III.2.2. Editeur de Code Java (Eclipse).....	28
III.2.2.1. Table de travail	29
III.2.2.2. Langue	30
III.2.2.3. Gestion des erreurs	30
III.2.3. Base de donnes MySQL	30
III.2.3.1. Fonctionnalités MySQL	Error! Bookmark not defined.
III.2.3.2. Langage SQL.....	31
III.3. Communication client/serveur TCP en JAVA	31
III.3.1. Côté serveur	32
III.3.2. Côté client.....	33
III.4. Transfert des chaines de caractère client /serveur	34
III.5. Transfert des objets entre Client - Serveur (Sérialisation d'objets).....	34
III.5.1. Définition.....	34
III.5.2. Objet sérialisable	35
III.6. Socket Multithread	36
III.6.1. Définition.....	36
III.6.2. Avantages du serveur multithread	36
III.6.3. Comment déclarer un thread.....	37
III.7. Application de gestion d'administration.....	38
III.7.1. Interfaces Coté Serveur	38
III.7.1.1. Connexion de l'administrateur (Serveur)	38
III.7.1.2. Gestion des étudiants	39
III.7.1.3. Gestion des enseignants.....	40
III.7.1.4. Gestion de la distribution de charges.....	40
III.7.1.5. Emploi du temps.....	41
III.7.2. Interface coté clients	41
III.7.2.1. Connexion de client	41
III.7.2.2. Création d'un compte étudiant et compte enseignant.....	42
III.7.2.3. Comptes enseignants	43
III.7.2.4. Distribution de charges	43
III.7.2.5. Comptes étudiants.....	44
III. 8. Conclusion	45
CONCLUSION GÉNÉRALE	46
REFERENCES BIBLIOGRAPHIQUES	47

LISTES DES FIGURES ET TABLEAUX

Figure I. 1 : Réseau informatique local et ses éléments.	4
Figure I. 2 : Protocoles de communication.....	5
Figure I. 3 : Réseau LAN.	7
Figure I. 4 : Réseau MAN.	8
Figure I. 5 : Réseau WAN.	9
Figure I. 6 : Réseau VPN.....	10
Figure I. 7 : Topologie en bus.	11
Figure I. 8 : Topologie en étoile.	11
Figure I. 9 : Topologie en anneau.....	12
Figure I. 10 : Topologie en arbre.....	13
Figure I. 11 : Topologie maillée.	13
Figure I. 12 : Couches du modèle OSI [3].....	14
Figure I. 13 : Comparaison entre OSI/ TCP-IP [3].	16
Figure II. 1 : Réseaux filaire, basés sur la technologie Ethernet [6].	20
Figure II. 2 : Exemple d'un réseau basé sur wifi [7].....	20
Figure II. 3 : Communication client serveur [7].....	22
Figure II. 4 : Type des communications client serveur [7].	24
Figure II. 5 : Communication entre client serveur a la base d'un socket [8].	26
Figure III. 1 : Interface principale de l'Eclipse.	29
Figure III. 2 : Comment créer un client et serveur simple.	32
Figure III. 3 : Sérialisation et désérialisation des objets [15].....	35
Figure III. 4 : Communication multithread [17].	36
Figure III. 5 : Connexion du serveur.....	38
Figure III. 6 : Page principale de l'administrateur.	39
Figure III. 7 : Gestion des comptes étudiants.....	39
Figure III. 8 : Gestion des comptes enseignants.	40
Figure III. 9 : Gestion distribution de charges.	40
Figure III. 10 : L'emploi du temps.....	41
Figure III. 11 : Création un compte étudiante.....	41
Figure III. 12 : Création un compte étudiant.....	42
Figure III. 13 : Page pour authentification des enseignants.	42
Figure III. 14 : Page principale pour les enseignants.	43
Figure III. 15 : Distribution de charges.....	44
Figure III. 16 : Page principale pour les étudiants.	44

LISTE DES TABLEAUX

Tableau I. 1 : Couches du modèle OSI.....	14
Tableau I. 2 : Espace d'adressage [4].....	18
Tableau III. 1 : Commendes principales de langage SQL [13].	31

ABREVIATIONS ET SYMBOLES

LAN	: Local Area Network
WLAN	: Wireless Local Area Network
OSI	: Organisation Internationale de Normalisation
HTTP	: HyperText Transfer Protocol
IP	: Internet Protocol
IEEE	: Institute of Electrical and Electronics Engineers
LTE	: Long Term Evolution
DSL	: Digital Subscriber Line
PAN	: Personal Area Network
MAN	: Metropolitan Area Network
WAN	: Wide Area Network
ATM	: Asynchronous Transfer Mode
GAN	: Generative Adversarial Network
VPN	: Virtual Private Network
FDDI	: Fiber Distributed Data Interface
TCP	: Transmission Control Protocol
UDP	: User Datagram Protocol
SGBD	: Système de Gestion de Base de Données
FTP	: File Transfer Protocol
DNS	: Domain Name System
Telnet	: Terminal Network
SQL	: Structured Query Language

INTRODUCTION GENERALE

Depuis plusieurs années, le monde assiste à de grands progrès notamment dans les moyens de gestion en termes de communication rapides, fiables et sécurisés. A cet effet, la plupart des pays espèrent avoir les dernières générations d'équipements assurant ce besoin. De nos jours, l'une des tendances les plus concernées c'est le développement des systèmes de télécommunication. Plus précisément les réseaux de télécommunication. Depuis leur apparition et introduction dans le monde de l'information et de la communication au sein du grand public et entreprises, ils assurent la liaison de différents coins du monde.

Un réseau de télécommunications peut être défini comme un ensemble d'équipements et de matériels informatiques reliés entre eux par des canaux de transmission et permettant le partage de ressources et l'échange de données. Selon l'organisation du réseau, l'architecture, la distance, la vitesse de transmission et la nature des informations transmises, le réseau doit suivre de nombreuses spécifications et normes [1]. Le réseau informatique est l'une des principales parties du réseau de télécommunications. Ce dernier est conçu pour connecter des équipements informatiques (serveurs, ordinateurs, imprimantes, etc.) pour échanger des données via des protocoles reposant sur la base d'un environnement client-serveur. Cependant comment réaliser un cet échange ou bien ce service en utilisant les protocoles de transport les plus connus (TCP, UDP) pour faire communiquer deux machines sur un réseau, c'est qu'on va voir dans ce mémoire.

Le travail présenté dans ce mémoire a principalement deux objectifs. Le premier est d'essayer d'apprendre les concepts de base des réseaux informatiques et le mode de communication entre client et serveur. Le second objectif consiste en la réalisation d'une application de gestion administrative de notre département, et pour y arriver, on a opté pour le langage Java, qui propose une architecture satisfaisante en termes de programmation.

Par ailleurs, on a divisé ce travail en trois chapitres. Le premier chapitre est consacré à présenter des généralités sur les notions fondamentales des réseaux informatiques, en particuliers de types (LAN, WAN, MAN, VPN ...etc.) et le modèle OSI.

Dans le second chapitre, on s'est intéressé essentiellement au modèle client-serveur qui présente la base de tous les services réseaux informatique, on a expliqué comment créer un réseau local LAN ensuite, on a présenté les différentes notions de base de ce modèle, les protocoles de transports utilisés, ainsi la notion de sockets, leur principe de fonctionnement et les protocoles utilisés.

Introduction générale

Le troisième chapitre présente les démarches suivies afin de réaliser l'application de gestion des étudiants et enseignants. Cette application offre la possibilité de relier les étudiants et leurs enseignants concernant la consultation des notes, planning et autres services.

Au final, on termine par une conclusion générale résumant notre travail et quelques perspectives sur l'amélioration de l'application de gestion proposée.

CHAPITRE I : NOTIONS FONDAMENTALES SUR LES RESEAUX INFORMATIQUES

I.1. Introduction

Dans nos jours, on vit dans un monde hyper actif où une multitude de technologies voient le jour chaque seconde grâce aux moyens informatiques mis à disposition, en particulier les réseaux informatiques, alors c'est quoi un réseau informatique ?

Chacun d'entre vous après avoir acheté un nouvel ordinateur, il est probable que la première chose qu'il fasse soit d'essayer de le connecter à Internet. Pour cela, vous allez établir une liaison avec le routeur, qui réceptionne les données venant d'Internet et les transmet à l'ordinateur. Mais ce n'est pas tout, vous aurez peut-être également envie de relier au routeur une imprimante, votre *Smartphone* ou votre téléviseur afin que ces appareils soient eux aussi connectés à Internet. Vous aurez donc mis en réseau plusieurs appareils via un point de commutation central et ainsi créé votre propre réseau.

I.2. Définition de réseau informatique

Dans les technologies de l'information, un réseau est défini par la mise en relation d'au moins deux systèmes informatiques au moyen d'un câble ou sans fil, par liaison radio. Le réseau le plus basique comporte deux ordinateurs reliés par un câble. On parle aussi dans ce cas de réseau *peer-to-peer* (P2P) ou en français pair à pair. Ce genre de réseau n'a pas de hiérarchie : les deux participants sont au même niveau. Chaque ordinateur a accès aux données de l'autre et ils peuvent partager des ressources, comme un disque de stockage, des programmes ou des périphériques (imprimante, etc.).

Les réseaux modernes sont un peu plus complexes en général et comportent bien plus que deux ordinateurs. Pour les systèmes à plus de dix participants, on utilise habituellement une configuration de type client/serveur. Dans ce modèle, un ordinateur agissant comme point de commutation central (serveur) met ses ressources à disposition des autres participants au réseau (clients).



Figure I. 1 : Réseau informatique local et ses éléments.

I.3. Principe de fonctionnement

Dans un réseau classique de type client/serveur, on trouve un point de commutation central, le serveur. Le serveur est connecté aux autres appareils, qui sont appelés clients. La connexion peut être filaire (LAN) ou sans fil (WLAN).

Dans un réseau domestique classique, le routeur joue le rôle de serveur. Il est relié à Internet et met la ressource « Internet » à disposition des autres appareils (ordinateur, *smartphone*, etc.). Le routeur rassemble tous les appareils, par connexion filaire et sans fil, au sein d'un réseau local.

I.4. Architecture client/serveur

Client : C'est le logiciel qui envoie des demandes à un serveur. Il peut s'agir d'un logiciel manipulé par une personne, ou d'un robot.

Serveur : Dépositaire centrale d'une fonction spécifique : service de base de données, de calcul, de fichier, mail, etc.

Dans les réseaux d'une certaine taille, par exemple en entreprise, le serveur est le plus souvent un ordinateur central. Cet ordinateur n'exécute pas des applications et des programmes

ordinaires, mais des logiciels et services de serveur spéciaux. Le serveur doit tourner en permanence, contrairement aux autres ordinateurs (*clients*), qui peuvent être éteints.

Dans ce genre de réseau basé sur serveur, la communication entre le serveur et le client s'effectue de la manière suivante : le client envoie d'abord une demande (*Request*) au serveur. Celui-ci examine la demande et renvoie ensuite les données (*Response*) voulues. Dans ce modèle, c'est toujours le client qui prend contact avec le serveur, jamais l'inverse.

I.4.1. Protocoles réseau

Les protocoles réseau assurent le bon déroulement de la communication entre les différents composants du réseau. Ils régulent l'échange de données, déterminent la façon dont la communication est amorcée et terminée et quelles données sont transférées. En règle générale, on trouve plusieurs protocoles réseau, qui assument chacun une sous-fonction donnée et sont disposés hiérarchiquement en couches.

Le schéma qui suit essaie de montrer où se situent divers protocoles dans le modèle OSI de l'ISO :

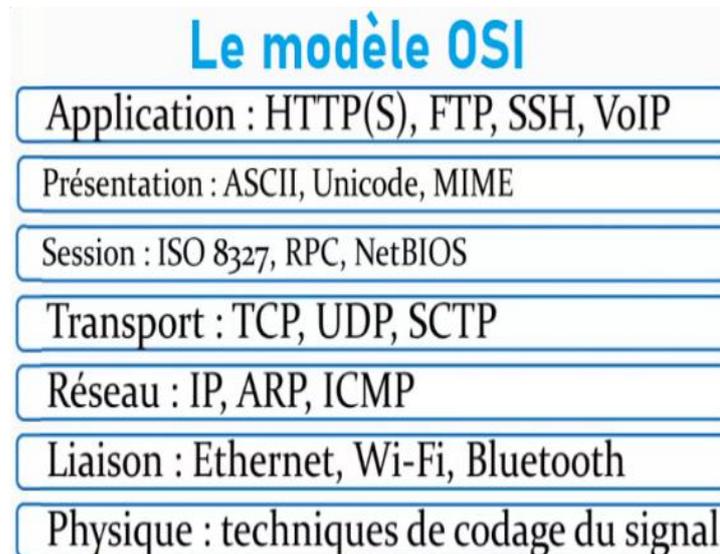


Figure I. 2 : Protocoles de communication.

I.4.2. Adresses réseau

À côté de cela, il convient de veiller à la bonne affectation de l'émetteur et du récepteur. À cette fin, on emploie ce que l'on appelle des adresses réseau. Au sein du réseau, chaque ordinateur possède généralement une adresse IP qui l'identifie de manière unique, à l'image d'un numéro de téléphone. Cette adresse IP interne sert uniquement à la communication entre les membres du réseau local. Pour communiquer sur le réseau Internet, on emploie à la place des adresses IP externes, qui sont attribuées automatiquement par les fournisseurs d'accès à Internet.

On distingue les adresses IPv4 et IPv6. Les adresses IPv4 étaient autrefois la norme, mais leur nombre était limité à environ 4,3 milliards. Suite à l'expansion massive d'Internet, il a fallu trouver en urgence des adresses IP supplémentaires. C'est dans cette optique qu'a été élaborée la nouvelle norme IPv6, qui permettra d'attribuer jusqu'à $3,4 \times 10^{38}$ adresses, autrement dit 340 sextillions, cela suffit pour longtemps.

I.5. Types de réseaux existents

Le plus souvent, on catégorise les réseaux en différents types de réseaux en fonction du mode de transmission des données et de leur portée.

I.5.1. Mode de transmission

Concernant le mode de transmission, on distingue les réseaux filaires et les réseaux sans fil. Les WLAN, qui reposent sur la norme IEEE 802.11 (appellation internationale : *Wi-Fi*), et la technologie LTE, utilisée par les téléphones mobiles et *smartphones*, sont des exemples de réseaux sans fil. Les réseaux filaires, comme DSL, sont également appelés « Internet haut débit ».

I.5.2. Portée

Pour la portée du réseau, on utilise les catégories suivantes :

I.5.2.1. Réseau Personnel (PAN)

Pour permettre l'échange de données des appareils modernes comme notamment les *smartphones*, tablettes, ordinateurs portables ou les ordinateurs de bureau, ces derniers peuvent être connectés à un réseau adapté. Celui-ci peut être relié sous la forme d'un réseau personnel ou PAN (*Personal Area Network*), on parle aussi de réseau domestique. Les techniques de transmission courantes sont l'USB ou le *FireWire*. Le réseau personnel sans fil (WPAN pour

Wireless Personal Area Network) repose sur des technologies comme le Bluetooth, USB sans fil, INSTEON, IrDA, ZigBee ou Z-Wave. Un réseau personnel sans fil qui peut être réalisé par l'intermédiaire du Bluetooth est appelé « Piconet ». Les WPAN et les PAN ne couvrent généralement que quelques mètres et ne sont pas adaptés pour connecter des appareils se trouvant dans des pièces ou bâtiments différents [1].

I.5.2.2. Réseaux locaux (LAN)

Si plusieurs ordinateurs doivent être réunis sur un réseau, cela se fait généralement sous la forme d'un réseau local ou LAN (acronyme de *Local Area Network*). Un tel réseau peut relier deux ordinateurs d'une maison ou alors plusieurs centaines d'appareils au sein d'une entreprise. Mais également des réseaux dans des institutions publiques comme les administrations, les écoles ou les universités sont généralement mis en œuvre sous la forme d'un LAN. Une norme commune très répandue pour les réseaux locaux câblés est le protocole Ethernet [1].

L'étendue géographique des réseaux locaux ne dépasse pas 10 km (ex. : pour un immeuble ou un campus). Le débit, ou la vitesse de communication, varie de quelques Mbps à 100 Mbps. Le nombre de stations ne dépasse généralement pas 1 000.

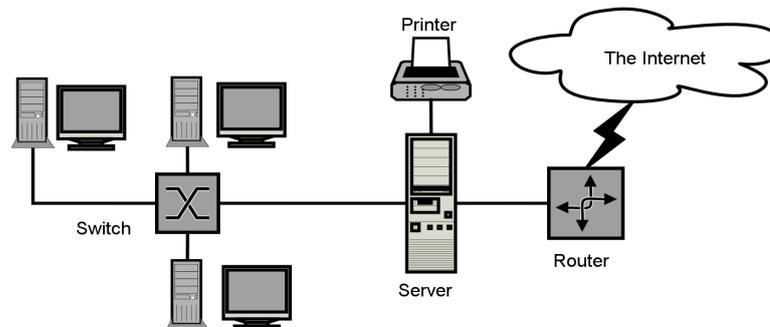


Figure I. 3 : Réseau LAN.

I.5.2.3. Réseaux Métropolitain (MAN)

Un réseau métropolitain ou *Metropolitan Area Network* (MAN), est un réseau permettant l'interconnexion de plusieurs réseaux locaux répartis sur différents sites dans une zone urbaine dont l'étendue géographique n'excède pas 200 km. Ces réseaux peuvent être privés ou publics. Le débit est élevé car supérieur à 100 Mbps (sur liens de fibre optique) [1].

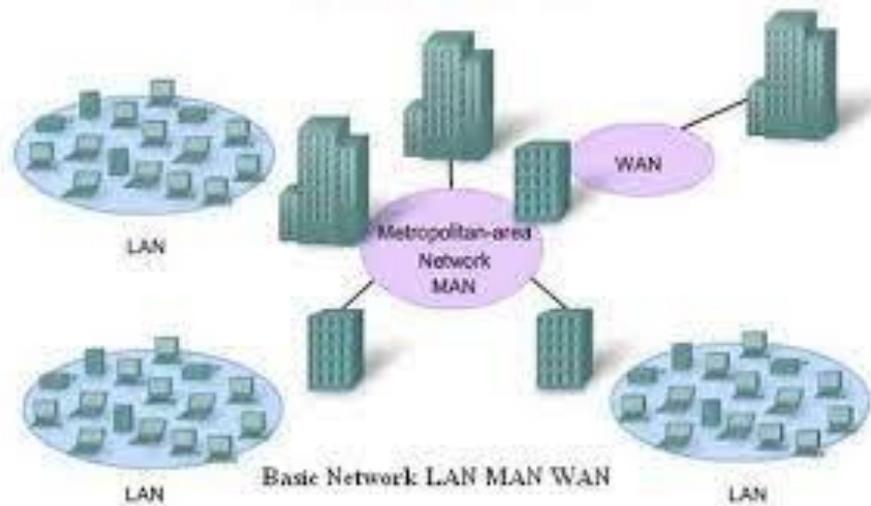


Figure I. 4 : Réseau MAN.

I.5.2.4. Réseaux Etendus (WAN)

Alors que les réseaux métropolitains relient des zones qui se trouvent proches les unes des autres dans des zones rurales ou urbaines, les WAN (*Wide Area Network*) ou réseaux étendus couvrent des vastes zones géographiques à l'échelle d'un pays ou d'un continent par exemple. En principe, le nombre de réseaux locaux ou d'ordinateurs connectés à un réseau étendu est illimité.

Alors que les réseaux locaux (LAN) et MAN peuvent être réalisés en raison de la proximité géographique des ordinateurs connectés ou des réseaux sur la base d'Ethernet, les réseaux étendus utilisent des techniques comme IP/MPLS (*Multi Protocol Label Switching*), PDH (*Plésiochrone Digitale Hiérarchie*), SDH (*Synchrone Digitale Hiérarchie*), SONET (*Synchrone Optical Network*), ATM (*Asynchrone Transfer Mode*) et encore rarement l'obsolète X25.

Les réseaux étendus sont généralement détenus par une organisation ou une entreprise et sont donc exploités en privé ou loués. En outre, les fournisseurs de services Internet utilisent des WAN pour connecter les réseaux locaux d'entreprises et les clients à Internet [1].

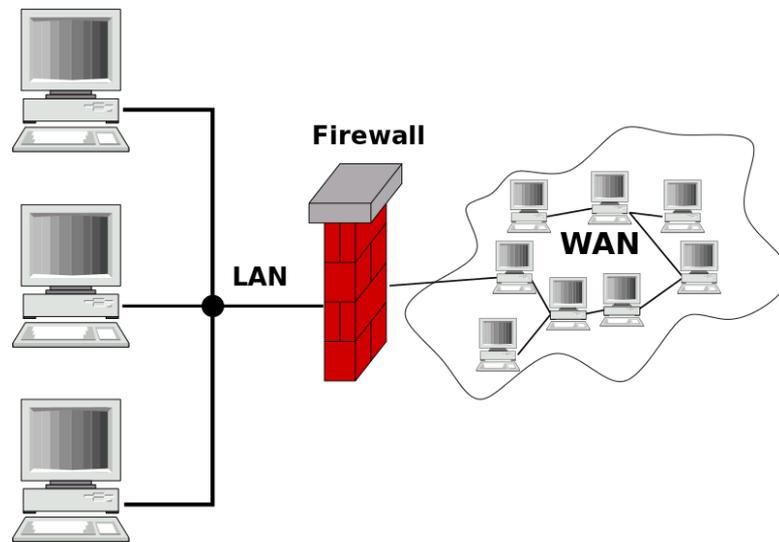


Figure I. 5 : Réseau WAN.

I.5.2.5. Réseaux Mondial (GAN)

Un réseau mondial comme Internet est aussi appelé GAN (*Globe Area Network*). Internet n'est cependant pas le seul réseau informatique de ce genre. Les entreprises actives au niveau international maintiennent également des réseaux isolés qui couvrent plusieurs WAN et connectent ainsi des ordinateurs d'entreprise dans le monde entier. Les GAN utilisent les infrastructures de fibre optique des réseaux étendus et combinent ces derniers avec des câbles sous-marins internationaux ou des transmissions par satellite [1].

I.5.2.6. Réseaux Privés Virtuels (VPN)

Un VPN (*Virtual Private Network*) ou réseau privé virtuel est un réseau de communication virtuel qui utilise l'infrastructure d'un réseau physique pour relier logiquement les systèmes informatiques. Il peut s'agir de n'importe quel type de réseau détaillé plus haut, cependant Internet est le moyen de transmission le plus couramment utilisé. Cela relie presque tous les ordinateurs dans le monde entier et reste disponible gratuitement par opposition à l'exploitation privé d'un MAN ou WAN. Les données sont transférées au sein d'un tunnel virtuel qui est construit entre un client VPN et un serveur VPN.

Le réseau public est utilisé comme moyen de transport, les réseaux privés virtuels sont généralement cryptés pour s'assurer de la confidentialité des données. Les VPN sont utilisés pour connecter les réseaux locaux sur Internet ou pour permettre l'accès à distance à un réseau ou à un seul ordinateur via la connexion publique [1].

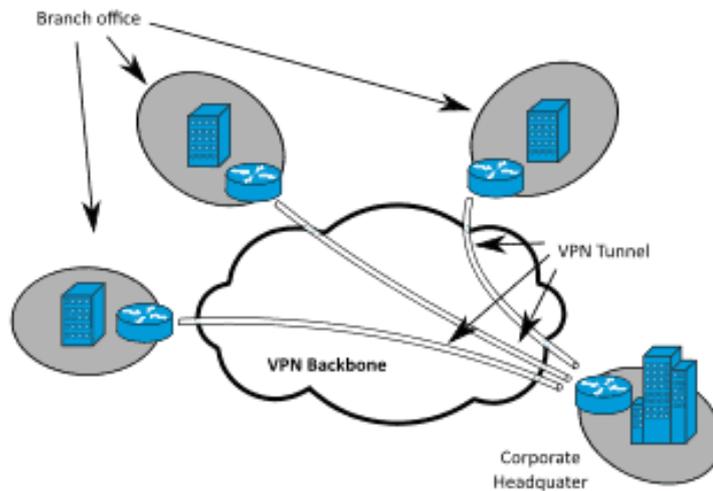


Figure I. 6 : Réseau VPN.

I.6. Topologie des réseaux

I.6.1. Définition de topologie

Un réseau informatique est constitué d'ordinateurs reliés entre eux grâce à des lignes de communication (câbles réseaux, etc.) et des éléments matériels (cartes réseau, ainsi que d'autres équipements permettant d'assurer la bonne circulation des données). L'arrangement physique, c'est-à-dire la configuration spatiale du réseau est appelée topologie physique. On distingue généralement les topologies suivantes :

- Topologie en bus
- Topologie en étoile
- Topologie en anneau
- Topologie en arbre
- Topologie maillée

La topologie logique, par opposition à la topologie physique, représente la façon dont les données transitent dans les lignes de communication. Les topologies logiques les plus courantes sont *Ethernet*, *Token Ring* et *FDDI* [2].

I.6.2. Topologie en bus

Une topologie en bus est l'organisation la plus simple d'un réseau. En effet, dans une topologie en bus tous les ordinateurs sont reliés à une même ligne de transmission par l'intermédiaire de câble, généralement coaxial. Le mot « bus » désigne la ligne physique qui relie les machines du réseau.

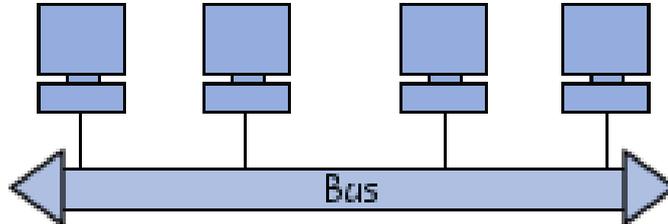


Figure I. 7 : Topologie en bus.

Cette topologie a pour avantage d'être facile à mettre en œuvre et de posséder un fonctionnement simple. En revanche, elle est extrêmement vulnérable étant donné que si l'une des connexions est défectueuse, l'ensemble du réseau en est affecté [2].

I.6.3. Topologie en étoile

Dans une topologie en étoile, les ordinateurs du réseau sont reliés à un système matériel central appelé concentrateur (en anglais *hub*, littéralement *moyen de roue*). Il s'agit d'une boîte comprenant un certain nombre de jonctions auxquelles il est possible de raccorder les câbles réseau en provenance des ordinateurs. Celui-ci a pour rôle d'assurer la communication entre les différentes jonctions.

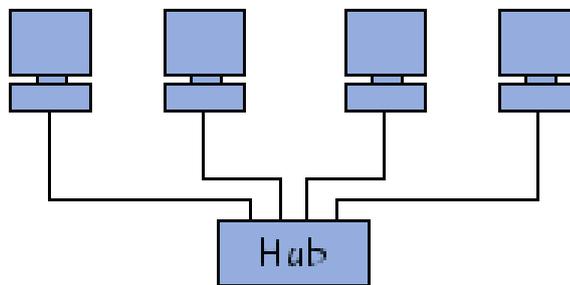


Figure I. 8 : Topologie en étoile.

Contrairement aux réseaux construits sur une topologie en bus, les réseaux suivant une topologie en étoile sont beaucoup moins vulnérables car une des connexions peut être débranchée sans paralyser le reste du réseau. Le point névralgique de ce réseau est le

concentrateur, car sans lui plus aucune communication entre les ordinateurs du réseau n'est possible. En revanche, un réseau à topologie en étoile est plus onéreux qu'un réseau à topologie en bus car un matériel supplémentaire est nécessaire (le hub) [2].

I.6.4. Topologie en anneau

Dans un réseau possédant une topologie en anneau, les ordinateurs sont situés sur une boucle et communiquent chacun à leur tour.

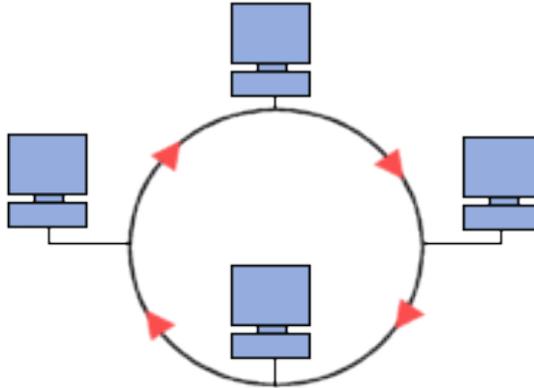


Figure I. 9 : Topologie en anneau.

En réalité, dans une topologie anneau, les ordinateurs ne sont pas reliés en boucle, mais sont reliés à un répartiteur (appelé *MAU*, *Multistation Access Unit*) qui va gérer la communication entre les ordinateurs qui lui sont reliés en impartissant à chacun d'entre-deux un temps de parole.

Les deux principales topologies logiques utilisant cette topologie physique sont *Token ring* (anneau à jeton) et FDDI [2].

I.6.5. Topologie en arbre

Aussi connu sous le nom de *topologie hiérarchique*, le réseau est divisé en niveaux. Le sommet, le haut niveau, est connectée à plusieurs nœuds de niveau inférieur, dans la hiérarchie. Ces nœuds peuvent être eux-mêmes connectés à plusieurs nœuds de niveau inférieur. Le tout dessine alors un arbre, ou une arborescence [2].

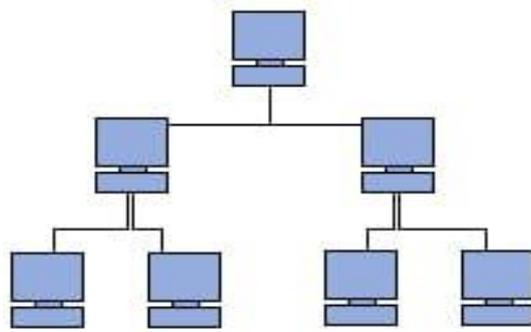


Figure I. 10 : Topologie en arbre.

I.6.6. Topologie maillée

Une topologie maillée, est une évolution de la topologie en étoile, elle correspond aux plusieurs liaisons point à point. Une unité réseau peut avoir (1, N) connexions point à point vers plusieurs autres unités. Chaque terminal est relié à tous les autres. L'inconvénient est le nombre de liaisons nécessaires qui devient très élevé. Cette topologie se rencontre dans les grands réseaux de distribution (Exemple : Internet). L'information peut parcourir le réseau suivant des itinéraires divers, sous le contrôle de puissants superviseurs de réseau, ou grâce à des méthodes de routage réparties. L'armée utilise également cette topologie, ainsi, en cas de rupture d'un lien, l'information peut quand même être acheminée [2].

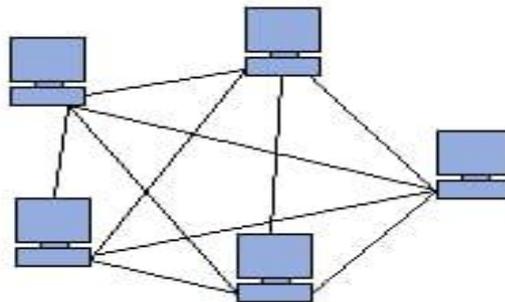


Figure I. 11 : Topologie maillée.

I.7. Modèle OSI (OPEN SYSTEMS INTERCONNECTION)

Le modèle OSI constitue un cadre de référence qui nous permet de comprendre comment les informations circulent dans un réseau. C’est aussi un modèle conceptuel d’architecture de réseau qui facilite la compréhension théorique du fonctionnement des réseaux. Il est constitué de sept couches, chacune définissant des fonctions particulières du réseau.

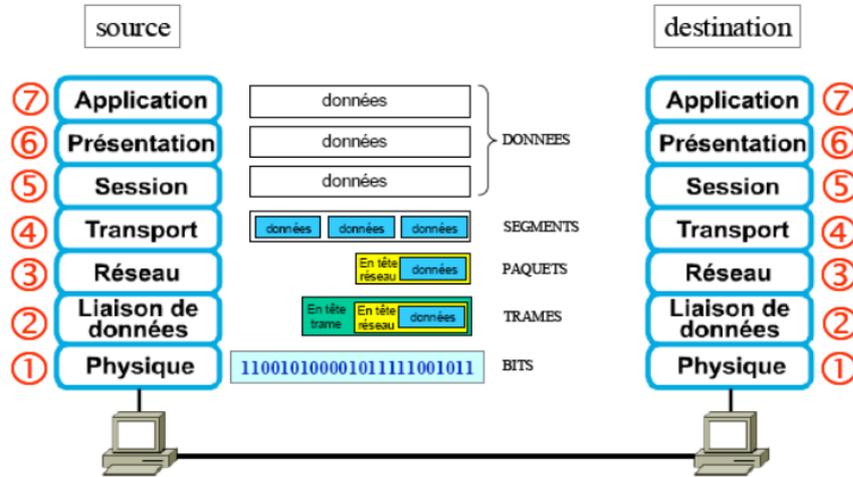


Figure I. 12 : Couches du modèle OSI [3].

7- Application	Cette couche est le point de contact entre l'utilisateur et le réseau. C'est donc elle qui va apporter à l'utilisateur les services de base offerts par le réseau, comme par exemple le transfert de fichier, la messagerie...
6- Présentation	Cette couche s'intéresse à la syntaxe et à la sémantique des données transmises : c'est elle qui traite l'information de manière à la rendre compatible entre tâches communicantes. Elle va assurer l'indépendance entre l'utilisateur et le transport de l'information. Typiquement, cette couche peut convertir les données, les reformater, les crypter et les compresser.
5- Session	La couche session fournit des services à la couche présentation pour organiser son dialogue et gérer l'échange de données.
4- Transport	Cette couche est responsable du bon acheminement des messages complets au destinataire. Le rôle principal de la couche transport est de prendre les messages de la couche session, de les découper s'il le faut en unités plus petites et de les passer à la couche réseau, tout en s'assurant que les morceaux arrivent correctement de l'autre côté. Cette couche effectue le réassemblage du message à la réception des morceaux.
3- Routage	La couche réseau fournit des services pour échanger les parties de données individuelles sur le réseau entre des périphériques terminaux identifiés.
2- Liaison	Les protocoles de la couche liaison de données décrivent des méthodes d'échanges de trames de données entre des périphériques sur un support commun.
1- Physique	Les protocoles de la couche physique décrivent les moyens mécaniques, électriques, fonctionnels et méthodologiques permettant d'activer, de gérer et de désactiver des connexions physiques pour la transmission de bits vers et depuis un périphérique réseau

Tableau I. 1 : Couches du modèle OSI.

I.8. Modèle TCP/IP (Transmission Control Protocol/Internet Protocol)

TCP/IP est l'architecture réseau la plus répandue. Le modèle est fondé sur OSI où 4 couches sont présentées : Application, Transport, Réseau et Liaison. L'architecture TCP/IP inclut de nombreux services d'application, des protocoles élaborés et complexes, sur lesquels les applications distribuées peuvent s'appuyer.

TCP/IP représente d'une certaine façon l'ensemble des règles de communication sur Internet et se base sur la notion d'adressage IP, c'est-à-dire le fait de fournir une adresse IP à chaque machine du réseau afin de pouvoir acheminer des paquets de données. Les fonctions essentielles du TCP/IP sont,

- Le fractionnement des messages en paquets.
- L'utilisation d'un système d'adresses.
- L'acheminement des données sur le réseau (routage).
- Le contrôle des erreurs de transmission de données.

I.9. Comparaison TCP/IP - OSI

La suite de protocoles désignée par TCP/IP, est construite sur un modèle en couches moins complet que la proposition de l'OSI. Quatre couches sont suffisantes pour définir l'architecture de ce protocole.

- Couche Application (*Application layer*).
- Couche Transport (Transport layer).
- Couche Internet (Internet layer).
- Couche interface réseau (Network access layer).
- Matériel (n'est pas une couche comprise dans le protocole) [4].

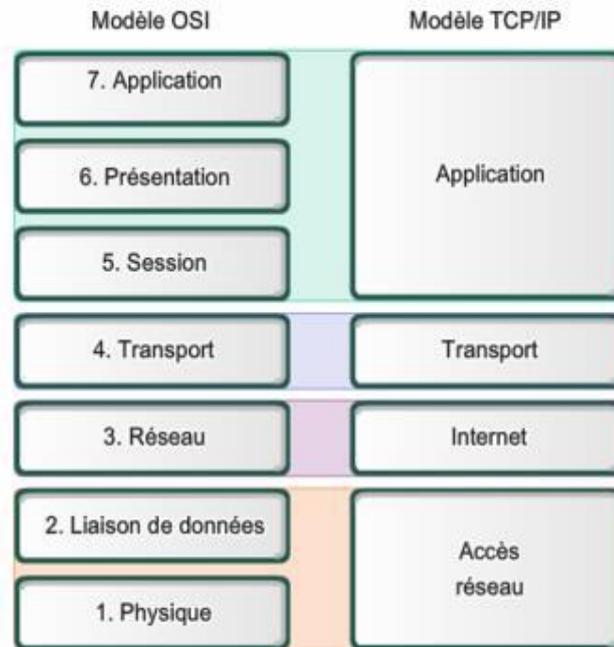


Figure I. 13 : Comparaison entre OSI/ TCP-IP [3].

I.10. Protocole IP (Internet Protocol)

Le protocole IP dans la couche réseau du TCP/IP se préoccupe de routage et de contrôle dans les réseaux interconnectés. Il permet également l'échange de datagrammes (en mode non connecté), entre des hôtes connectés à des réseaux physiques divers. Le protocole IP est un protocole ouvert et indépendant du système.

Des nœuds d'un réseau sont potentiellement reliés par plusieurs chemins. Chaque paquet suit un chemin particulier pour atteindre sa destination, mais une succession de paquets provenant d'un même bloc de données ne suivent pas forcément le même chemin. Pour programmer via le protocole IP on utilise le TCP ou UDP.

I.11. Adressage IPV4

Chaque équipement, que ce soit une station de travail, un routeur ou un serveur, doit avoir une adresse IP unique. Cette adresse ne dépend pas du matériel utilisé pour relier les machines ensemble, c'est une adresse logique notée sous forme de : w.x.y.z

Exemple d'adresses IP : 212.217.0.12

I.11. 1. Anatomie d'une adresse IP

Une adresse IP est un nombre de 32 bits codé sur 4 octets (octet = 8 bits) séparés par un point. On trouve souvent cette adresse avec des valeurs décimales. On appelle cette notation le décimal pointé.

Mais il est possible de l'écrire sous forme binaire :

Exemple : L'adresse IP 212.217.0.1 correspond à la notation binaire :

11010100.11011001.00000000. 00000001

Chaque nombre est compris entre 0 et 255, soit en binaire entre 00000000 et 11111111.

Toute adresse IP est composée de deux parties distinctes :

- Une partie nommée Identificateur (ID) du réseau : net-ID située à gauche, elle désigne le réseau contenant les ordinateurs.
- Une autre partie nommée identificateur de l'hôte : host-ID située à droite et désignant les ordinateurs de ce réseau.

I.11.2. Masque de réseau

Pour que le réseau Internet puisse router (acheminer) les paquets de données, il faut qu'il connaisse l'adresse du réseau de destination. Pour déterminer cette adresse réseau à partir de l'adresse IP de destination, on utilise le masque de sous réseau.

Le masque de réseau, ou *Net Mask*, est constitué de 32 bits. Les bits à « 1 » sont tous à gauche alors que les « 0 » sont tous à droite. On dit que les bits à « 1 » sont contigus (c'est-à-dire collés).

Exemples de masques :

11111111.00000000.00000000.00000000 = 255.0.0.0

11111111.11111111.11111111.00000000 = 255.255.255.0

11110000.00000000.00000000.00000000 = 240.0.0.0

Exemple de masque invalide :

11111111.01111111.00000000.00000000

Parmi les adresses possibles, deux sont spécifiques et ne doivent pas être utilisées par des machines :

Tous les bits de la partie Host-ID sont à 0 : C'est l'adresse du réseau

Ex : $192.168.10.0 / 255.255.255.0 = 192.168.10.00000000$

Tous les bits de la partie Host-ID sont à 1 : C'est l'adresse de diffusion utilisée pour communiquer avec toutes les machines du réseau.

Ex : $172.27.255.255 / 255.255.0.0 = 172.27.11111111.11111111$

I.11.3. Classes d'adresses

Les classes d'adresses réseaux correspondent à des regroupements en réseaux de même taille. Les réseaux de la même classe ont le même nombre d'hôtes maximum.

Classe	Masque réseau	Adresses réseau	Nombre de réseau	Nombre d'hôtes par réseau
A	255.0.0.0	1.0.0.0 - 126.255.255.255	126	16777214
B	255.255.0.0	128.0.0.0 - 191.255.255.255	16384	65534
C	255.255.255.0	192.0.0.0 - 223.255.255.255	2097152	254
D	240.0.0.0	224.0.0.0 - 239.255.255.255	Adresse unique	Adresse unique
E	Non défini	240.0.0.0 - 255.255.255.255	Adresse unique	Adresse unique

Tableau I. 2 : Espace d'adressage [4].

Adresses de classe A

Les adresses de classe A ont une partie réseau sur 8 bits, et une partie hôte sur 24 bits. Leur bit de poids le plus fort est 0, ce qui permet de les distinguer des autres classes.

Adresses de classe B

Les adresses de classe B ont une partie réseau sur 16 bits, et une partie hôte de même taille.

Adresses de classe C

Les adresses de classe C ont une partie réseau sur 24 bits, et une partie hôte sur 8 bits. Leurs trois bits de poids fort sont 1 1 0, ce qui permet de les distinguer des autres classes deux bits de poids forts sont 1 0, ce qui permet de les distinguer des autres classes.

I.12. Conclusion

Dans ce premier chapitre, on a tenté de balayer un peu les concepts de base du réseau informatique à savoir son architecture, ses protocoles et comment fonctionne les types de réseaux existents (LAN, WAN, VPN).

On a également donné un aperçu sur l'OSI qui est un modèle conceptuel d'architecture de réseau qui facilite la compréhension théorique du fonctionnement des réseaux, et au final on vu le protocole ouvert et indépendant du système appelé IP qui permet l'échange de datagrammes entre des hôtes connectés à des réseaux physiques divers.

CHAPITRE II : Réseau local LAN et ses protocoles

II.1. Introduction

La communication entre ordinateurs est similaire à ce qui est utilisé entre humains. Lorsque l'on communique, un expéditeur envoie un message à un destinataire en utilisant un support. Ils doivent se mettre d'accord sur le langage et sur la manière de décrire le message. Donc comment communiquent les applications sur différentes machines, et quels sont les protocoles utilisés dans ces communications ?

II.2. Création d'un réseau local (LAN)

Lorsque vous disposez de plusieurs ordinateurs, il peut être pratique de les connecter ensemble afin de créer un réseau local (LAN, abréviation de *Local Area Network*). La mise en place d'un tel réseau est très peu onéreuse, contrairement à ce que l'on pourrait croire. Voici quelques avantages que peut procurer la mise en place d'un réseau local :

- Transfert de fichiers ;
- Partage de ressources (partage de la connexion à internet, partage d'imprimante, disques partagés, etc.) ;
- Mobilité (dans le cas d'un réseau sans fil) ;
- Discussion (essentiellement lorsque les ordinateurs sont distants) ;
- Jeu en réseau.

II.2.1. Deux types de réseaux locaux

Il existe deux principaux types d'architecture de réseau local [5] :

- ✚ **Réseaux filaire, basés sur la technologie Ethernet** : représentant la quasi-totalité des réseaux locaux. Etant donné que les réseaux Ethernet utilisent généralement des câbles RJ45, on parle souvent de réseaux RJ45 ;

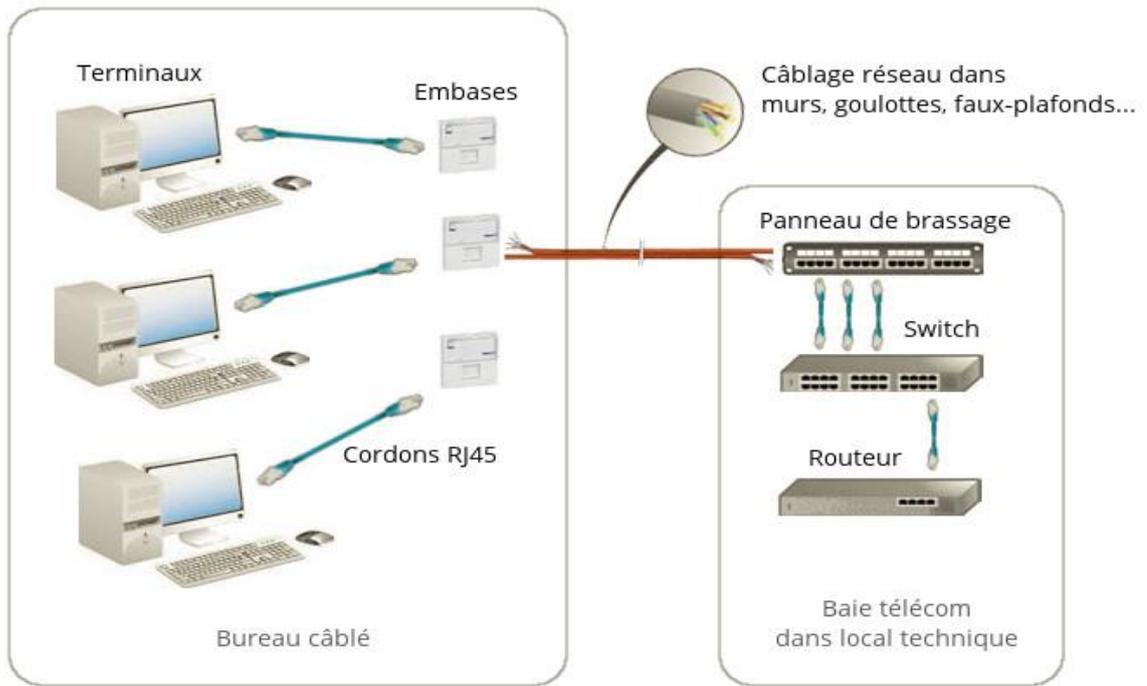


Figure II. 1 : Réseaux filaire, basés sur la technologie Ethernet [6].

✚ Réseaux sans fil : Utilisant généralement la technologie Wifi.



Figure II. 2 : Exemple d'un réseau basé sur wifi [7].

II.2.2. Matériels nécessaires

Pour créer un réseau local en RJ45 sous Windows, il suffit de :

- Plusieurs ordinateurs tournant sous Windows (des ordinateurs tournant sous deux versions différentes de Windows peuvent faire partie du même réseau) ;
- Des cartes Ethernet, sur port PCI (comportant une prise RJ45) ou intégrées à la carte mère. Vérifiez le cas échéant que les diodes à l'arrière de la carte réseau s'allument lorsque l'ordinateur est sous-tension et qu'un câble est branché ! Il existe également des adaptateurs réseau sur port USB, notamment dans le cas d'adaptateurs réseau sans fil ;
- Des câbles RJ45, dans le cas de réseaux filaires ;
- Un concentrateur (également appelé *hub*), boîtier auquel il est possible de connecter les câbles RJ45 provenant des différents ordinateurs du réseau, ou un commutateur (*Switch*) au lieu du hub, si vous souhaitez connecter uniquement deux ordinateurs il est préférable d'utiliser un câble croisé.

II.2.3. Comment envoyer des messages entre machines dans un réseau local ?

Chaque machine du réseau possède une adresse. Après avoir envoyé un message sur le réseau local en utilisant la couche 1, chaque machine recevant le message vérifie si elle est destinataire du message en comparant l'adresse de destination avec sa propre adresse.

Le protocole utilisé constitue la couche 2 (liaison) de la pile de communication. Ex : le protocole Ethernet.

II.2.4. Comment envoyer des messages entre machines dans deux réseaux différents ?

Une machine particulière (routeur) permet de relier différents réseaux. Un protocole se charge de déterminer une route (un chemin) permettant de relier les 2 machines distantes. Le protocole utilisé constitue la couche 3 (réseau) de la pile de communication.

Le protocole IP où les machines ont une adresse et le protocole se charge au mieux de l'acheminement des paquets jusqu'à destination.

II.2.5. Comment communiquent des applications sur différentes machines ?

On envoie le message dans un paquet IP utilisant l'adresse IP de la machine destinataire. Problème : la machine ne saura pas à quelle application est destiné le message. On donne une

adresse aux applications et on utilise le protocole de la couche 3 pour envoyer le message, en indiquant en plus l'adresse de l'application à exécuter dans la machine destinataire. Le protocole gère aussi parfois les connections offrant certaines garanties sur la transmission des messages [5].

Le protocole utilisé constitue la couche 4 (transport) de la pile de communication. Les adresses des applications dans la machine sont appelées des ports. Ex : les protocoles TCP ou UDP.

II.3. Modèle client-serveur

II.3.1. Définition

Le modèle client-serveur s'articule autour d'un réseau auquel sont connectés deux types d'ordinateurs le serveur et le client. Le client et le serveur communiquent via des protocoles. Les applications et les données sont réparties entre le client et le serveur de manière à réduire les coûts. Le client-serveur représente un dialogue entre deux processus informatiques par l'intermédiaire d'un échange de messages. Le processus client sous-traite au processus serveur des services à réaliser. Les processus sont généralement exécutés sur des machines, des OS et des réseaux hétérogènes [5].

Client : C'est le logiciel qui envoie des demandes à un serveur. Il peut s'agir d'un logiciel manipulé par une personne, ou d'un robot.

Serveur : Dépositaire centrale d'une fonction spécifique : service de base de données, de calcul, de fichier, mail, etc.

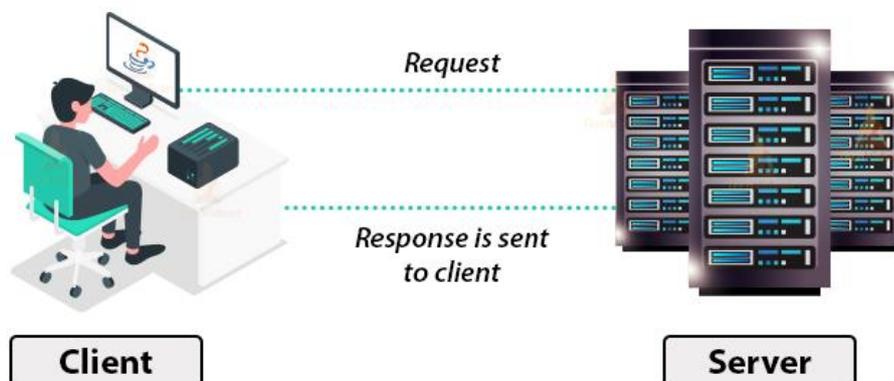


Figure II. 3 : Communication client-serveur [7].

II.3.2. Caractéristiques des systèmes client-serveur

Les éléments qui caractérisent une architecture client-serveur sont :

II.3.2.1. Service

Le modèle client-serveur est une relation entre des processus qui tournent sur des machines séparées. Le serveur est un fournisseur de services. Le client est un consommateur de services [5].

II.3.2.2. Partage de ressources

Un serveur traite plusieurs clients et contrôle leurs accès aux ressources

II.3.2.3. Protocole asymétrique

Conséquence du partage de ressources, le protocole de communication est asymétrique le client déclenche le dialogue ; le serveur attend les requêtes des clients [5].

II.3.2.4. Transparence de la localisation

L'architecture client-serveur doit masquer au client la localisation du serveur (que le service soit sur la même machine ou accessible par le réseau). Transparence par rapport aux systèmes d'exploitation et aux plates-formes matérielles. Idéalement, le logiciel client serveur doit être indépendant de ces deux éléments [5].

II.3.2.5. Message

Les messages sont les moyens d'échanges entre client et serveur.

II.3.2.6. Encapsulation des services

Un client demande un service. Le serveur décide de la façon de le rendre une mise à niveau du logiciel serveur doit être sans conséquence pour le client tant que l'interface message est identique [5].

II.3.2.7. Evolution

Une architecture client-serveur doit pouvoir évoluer horizontalement (évolution du nombre de clients) et verticalement (évolution du nombre et des caractéristiques des serveurs) [5].

II.3.3. Types de modèle client serveur

En fait, les différences sont essentiellement liées aux services qui sont assurés par le serveur. On distingue couramment :

II.3.3.1. Client -serveur de donnée

Dans ce cas, le serveur assure des tâches de gestion, stockage et de traitement de donnée. C'est le cas le plus connu de client- serveur qui est utilisé par tous les grands SGBD,

La base de données avec tous ses outils (maintenance, sauvegarde...) est installée sur un poste serveur. Pour les clients, un logiciel d'accès est installé permettant d'accéder à la base de données du serveur, tous les traitements sur les données sont effectués sur le serveur qui renvoie les informations demandées par le client [5].

II.3.3.2. Client -serveur de présentation

Dans ce cas la présentation des pages affichées par le client est intégralement prise en charge par le serveur. Cette organisation présente l'inconvénient de générer un fort trafic réseaux [5].

II.3.3.3. Client –serveur de traitement

Dans ce cas, le serveur effectue des traitements à la demande du client. Il peut s'agir de traitement particulier sur des données, de vérification de formulaire de saisie, de traitements d'alarmes. Ces traitements peuvent être réalisés par des programmes installés sur des serveurs mais également intégrés dans des bases de données, dans ce cas, la partie donnée et traitement sont intégrés [5].

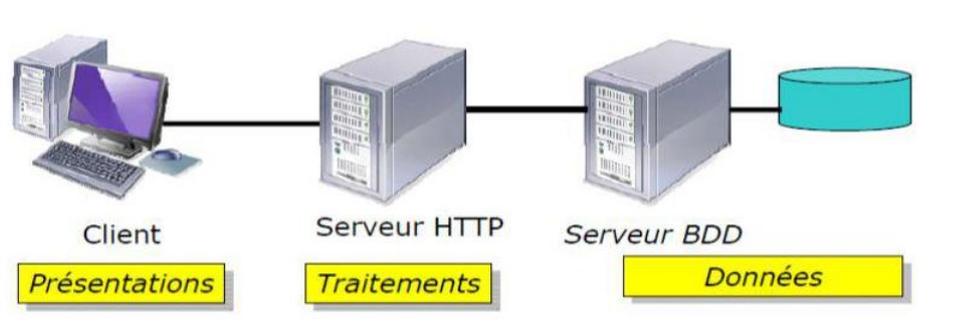


Figure II. 4 : Type des communications client-serveur [7].

II.4. Notion de port

Lors d'une communication en réseau, les différents ordinateurs s'échangent des informations qui sont généralement destinées à plusieurs applications. Seulement ces informations transitent par la même passerelle. Il faut donc savoir pour quelle application telle information est destinée. On attribue donc des ports pour chaque application. Un port est comme une porte en schématisant. Les informations sont multiplexées (comme dans les voitures récentes) et passent par la passerelle. A leur arrivée (vers le serveur) ou à leur réception (vers votre machine) elles sont dé-multiplexées et chaque information distincte passe par le port qui lui est associé. Les informations sont ensuite traitées par l'application correspondante.

Un port est codé sur 16 bits, il y a donc 65536 ports. L'adresse IP plus le port soit exemple : 127.0.0.1:80) est appelée **socket**. Les ports se sont vus attribuer une assignation par défaut pour aider à la configuration des réseaux.

Voici les principaux ports et protocoles les plus utilisés : Port Service ou Application.

Exemple : (FTP21), (23Telnet), (MTP 53), (DNS 80), (http 110).

Les ports de 0 à 1023 sont les ports reconnus ou réservés et sont assignés par l'IANA (*Internet Assigned Numbers Authority*).

Les ports de 1024 à 49151 sont appelés ports enregistrés et les ports 49152 à 65535 sont les ports dynamiques.

II.5. Socket

II.5.1. Historique

L'utilisation du terme "Socket" a commencé en 1971 où il a été utilisé lors du développement d'ARPANET. La plupart des prises mises en œuvre aujourd'hui sont basées sur les prises Berkeley, qui ont été développées en 1983. Les prises utilisées pour établir une connexion Internet sont toutefois calquées sur les modèles *Winsock* fabriqués en 1991. Les prises Berkeley sont également connues sous le nom de prise BSD. En 1989, Berkeley a publié des versions de son système d'exploitation et de sa bibliothèque réseau sans contraintes de licence. D'autres premières implémentations ont été écrites pour TOPS-20, MVS, VM et IBM-DOS.

II.5.2. Définition Sockets

Un socket est connu comme un type de logiciel qui agit comme un point d'extrémité qui fonctionne en établissant une liaison de communication réseau bidirectionnelle entre l'extrémité du serveur et le programme de réception du client. On l'appelle aussi souvent un point d'aboutissement dans un canal de communication bidirectionnel. Ces sockets sont réalisés et mobilisés en même temps qu'un ensemble de requêtes de programmation identifiées comme appels de fonction, qui est techniquement appelé interface de programmation d'application (API). Un socket est capable de simplifier le fonctionnement d'un programme car les programmeurs n'ont plus qu'à se soucier de manipuler les fonctions du socket, ce qui leur permet de compter sur le système d'exploitation pour transporter correctement les messages sur le réseau [7].

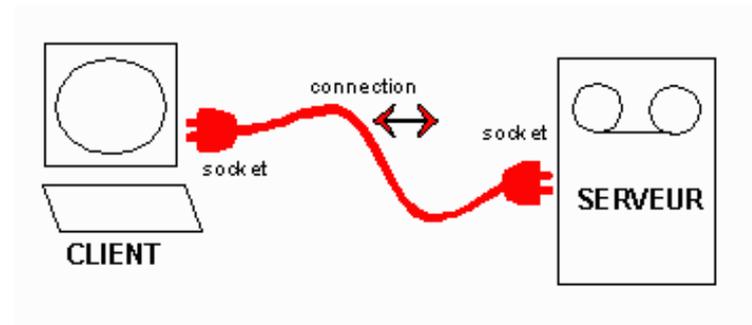


Figure II. 5 : Communication entre client-serveur à la base d'un socket [8].

II.5.3. Principe de fonctionnement

Typiquement, un socket respecte un flux spécifique d'événements pour qu'elle fonctionne normalement. Pour un modèle client-serveur orienté connexion, le socket du processus serveur attend la demande d'un client. Pour ce faire, le serveur doit d'abord établir une adresse que les clients peuvent utiliser pour trouver et se connecter au serveur. Lorsqu'une connexion est établie avec succès, le serveur attend que les clients demandent un service. L'échange de données client-serveur aura lieu si le client se connecte au serveur via le socket. Le serveur répondra alors à la demande du client et lui enverra une réponse.

La plupart du temps, les URL et leurs connexions sont utilisées pour accéder à Internet. Les programmes nécessiteront un simple lien de communication entre le client et le serveur du programme. Ce rôle est associé à un socket qui aidera à lier les extrémités client et serveur du programme. Si un client commence à établir une communication avec le serveur, une connexion fiable entre le serveur et le client sera établie par le canal de communication TCP. Avec ce type de communication, le client et le serveur peuvent lire ou écrire sur des sockets qui sont liés à un canal de communication spécifique [9].

II.6. Protocoles utilisés dans sockets

Vous pouvez créer des transports Socket qui facilitent la communication entre les clients et les serveurs. Auparavant, vous pouvez étudier comment *Intégration Tester* fournit l'accès aux couches OSI application et transport.

La couche transport surveille la livraison de données entre deux processus, l'un situé sur un ordinateur et l'autre, sur un ordinateur différent. Les protocoles de la couche transport servent de liaisons entre les protocoles de la couche application et les services fournis par le réseau. Deux protocoles de la couche transport sont pris en charge par *Intégration Tester* : UDP et TCP.

II.6.1. Protocole de datagramme utilisateur (UDP)

UDP (*User Datagramme Protocol*) est un protocole de transport sans connexion (et par conséquent, peu fiable). Il exécute peu de contrôle d'erreur et n'ajoute rien aux services IP (*Internet Protocol*), excepté pour fournir une communication entre processus plutôt qu'entre hôtes.

UDP est un protocole simple consommant un temps système minimal. Si un processus souhaite envoyer un petit message en se souciant peu de la fiabilité, il peut utiliser UDP. L'envoi d'un message via UDP prend nettement moins de temps qu'avec TCP [9].

II.6.2. Protocole TCP

TCP (*Transmission Control Protocol*) est un protocole de la couche transport fiable mais complexe. TCP ajoute des fonctions orientées connexion et la fiabilité à IP.

TCP est un service fiable qui garantit la livraison d'un flux de données entre deux hôtes, sans duplication ni perte de données. Etant donné que le transfert de paquets n'est pas fiable, une technique désignée "accusé de réception positif avec retransmission" est utilisée pour garantir la fiabilité des transferts de paquets. Cette technique fondamentale exige que le récepteur réponde par un message d'accusé de réception lorsque qu'il reçoit les données [9].

L'émetteur conserve un enregistrement de chaque paquet qu'il envoie et attend l'accusé de réception avant d'envoyer le paquet suivant. L'émetteur déclenche également un temporisateur lors de l'envoi de données et retransmet le paquet si le temporisateur expire. Le temporisateur est nécessaire en cas de perte ou d'endommagement d'un paquet [9].

II.7. Conclusion

Dans ce chapitre on s'est intéressé essentiellement au modèle client-serveur qui est la base de tous les services réseaux informatique, on a expliqué comment créer un réseau local LAN, ensuite on a présenté les différentes notions de base de ce modèle, les protocoles de transports utilisés, ainsi la notion de sockets, leur principe de fonctionnement et les protocoles utilisés.

CHAPITRE III : *Application de gestion des étudiants et des enseignants en Java*

III.1. Introduction

Dans nos jours, les applications font partie intégrante de notre vie quotidienne, voir même une nécessité primordiale notamment au sein des différents structures, à cet effet ce chapitre présente l'essentiel de notre travail qui se résume dans la création d'une application en Java développée dans le cadre de ce projet.

Dans un premier temps, nous allons présenter les langages utilisés à savoir le langage JAVA et le langage de gestion de base de données SQL, ensuite nous allons exposer l'organigramme qui résume la communication client-serveur TCP, le transfert de données en utilisant les objets et la notion des serveurs multi threads. La deuxième partie sera consacrée à la présentation de notre application.

III.2. Langages Utilisés

III.2.1. Langage JAVA

Depuis son origine, Java fournit plusieurs classes et interfaces destinées à faciliter l'utilisation du réseau par programmation en reposant sur les sockets. Celles-ci peuvent être mises en œuvre pour réaliser des échanges utilisant le protocole réseau IP avec les protocoles de transport TCP ou UDP.

III.2.2. Editeur de Code Java (Eclipse)

Plusieurs éditeurs de code JAVA ont été développés afin d'assurer une fiabilité de code, nous citons : NetBeans, JDeveloper, JCreator, Xcode, Eclipse, etc. Dans cette partie nous allons exposer l'un de ces éditeurs que nous avons utilisés dans ce projet.

Eclipse est une plate-forme d'outils universelle et un environnement de développement intégré ouvert et extensible qui peut être utilisé pour réaliser beaucoup de choses. Sa vraie valeur vient des plug-ins d'outils, qui peuvent "apprendre" à Eclipse à travailler avec d'autres fichiers Java, contenu Web, images, vidéos...etc. Eclipse vous permet de développer indépendamment des outils logiciels qui peuvent être intégrés de manière transparente avec d'autres outils, et vous n'avez pas besoin de savoir où se termine un outil et où commence un autre [10].

III.2.2.1. Table de travail

La table de travail est une collection de plusieurs fenêtres. Chaque fenêtre contient une barre de menus, une barre d'outils, une barre de raccourcis et une ou plusieurs perspectives [10].

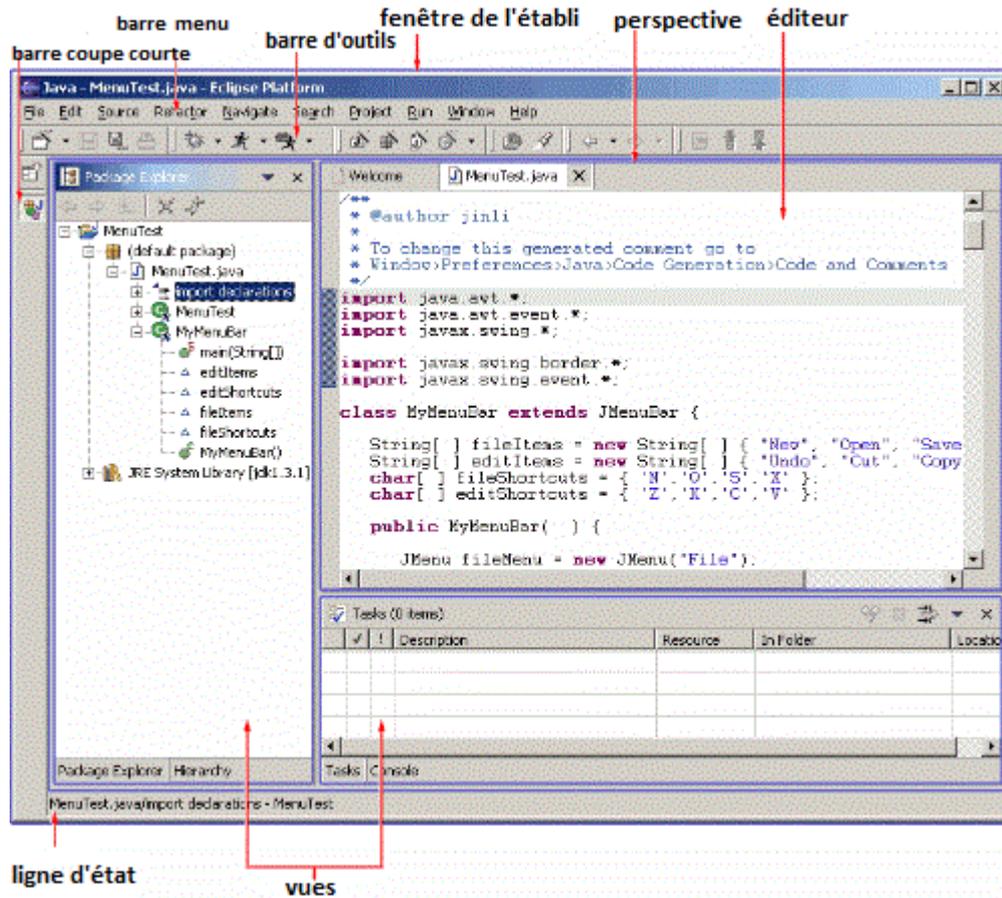


Figure III. 1 : Interface principale de l'Eclipse.

Une vue en perspective est un conteneur visuel qui contient une série de vues et d'éditeurs de contenu. Les vues sont entièrement dans une perspective et ne peuvent pas être partagées, et tout éditeur de contenu ouvert peut être partagé entre les perspectives. Si deux perspectives ou plus ouvrent la même vue, elles partagent la même instance de cette vue, même si la disposition des vues peut être différente entre les différentes perspectives. Pour la vue en perspective dans différentes fenêtres de table de travail, ni l'éditeur ni la vue ne peuvent être partagés. La perspective de visualisation est comme une page dans un livre. C'est dans une fenêtre et avec d'autres perspectives, tout comme une page dans un livre, vous ne pouvez voir qu'une perspective à la fois.

La barre de menus principale de table de travail comprend généralement des menus de niveau supérieur tels que Fichier, Modifier, Naviguer, Projet, Fenêtre et Aide. D'autres menus

de niveau supérieur sont situés entre les menus Edition et Projet et sont souvent contextuels. Ce contexte comprend la vue de l'activité en cours, l'éditeur frontal (qu'il soit actif ou non) et la vue active [10].

III.2.2.2. Langue

Eclipse peut être utilisé sur différentes plates-formes et dans différentes régions. Pour prendre en charge différentes langues et formats numériques, des stratégies de localisation doivent être appliquées au texte et aux images de chaque plug-in. Cela implique de séparer toutes les ressources du code source du plugin afin que ces ressources puissent être converties pour s'adapter à la nouvelle région.

III.2.2.3. Gestion des erreurs

Si une erreur se produit dans Eclipse, une réponse appropriée doit être apportée en fonction du contexte de l'erreur,

- Veuillez-vous référer à notre guide pour une partie de la façon de gérer les erreurs de saisie de l'utilisateur dans l'assistant.
- Veuillez-vous référer à l'Éditeur voire une partie du guide sur la façon de gérer les erreurs dans l'Éditeur.

Si l'erreur nécessite une saisie explicite ou une attention immédiate de la part de l'utilisateur, une boîte de dialogue modale doit être utilisée pour informer l'utilisateur de l'erreur. Cela peut obliger l'utilisateur à faire attention et à traiter le problème immédiatement.

III.2.3. Base de données

Une base de données une entité dans laquelle il est possible de stocker des données de façon structurée et avec le moins de redondance possible. Ces données doivent pouvoir être utilisées par des programmes, par des utilisateurs différents. Ainsi, la notion de base de données est généralement couplée à celle de réseau, afin de pouvoir mettre en commun ces informations, d'où le nom de **base**. On parle généralement de système d'information pour désigner toute la structure regroupant les moyens mis en place pour pouvoir partager des données.

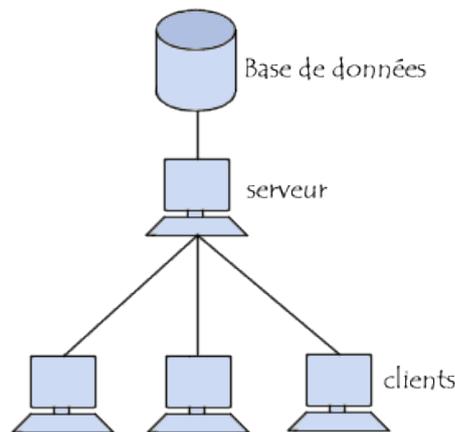


Figure III.2. Base de données SQL[12].

III.2.3.2. Langage SQL

C'est un langage permettant la communication avec une base de données. Ce langage informatique est notamment très utilisé par les développeurs web pour communiquer avec les données d'un site web. [13].

Le tableau suivant résume les principales commandes SQL :
SELECT, INSERT INTO, UPDATE, DELETE, DROP TABLE

Instructions SQL	Explications
SELECT nom, prénom FROM tb_salaries	Extrait les valeurs de tous les champs de toutes les personnes présentes dans la table « tb_salaries ».
SELECT nom, prénom FROM tb_salaries WHERE date_sortie is null	Extrait les valeurs des champs « nom » et « prénom » de toutes les personnes présentes dans la table « tb_salaries » (ceux dont la date de sortie est nulle).
INSERT INTO tb_salaries («ARMSTRONG», «Neil», «08/05/30»)	Ajoute le salarié « Neil Armstrong » avec sa date de naissance dans la table « tb_salarie ». Les champs n'étant pas spécifiés, ils doivent forcément être donnés dans l'ordre de la structure de la table.
UPDATE tb_salaries SET date_naissance = « 08/04/1901 » WHERE nom= «ARMSTRONG » AND prénom= »Louis »	Mis à jour de la date de naissance de Louis Armstrong,

Tableau III. 1 : Commandes principales de langage SQL [13].

III.3. Communication client/serveur TCP en JAVA

Dans cette partie nous allons décrire les étapes nécessaires pour réaliser une communication client-serveur TCP en utilisant le langage JAVA. La figure suivante résume ces étapes :

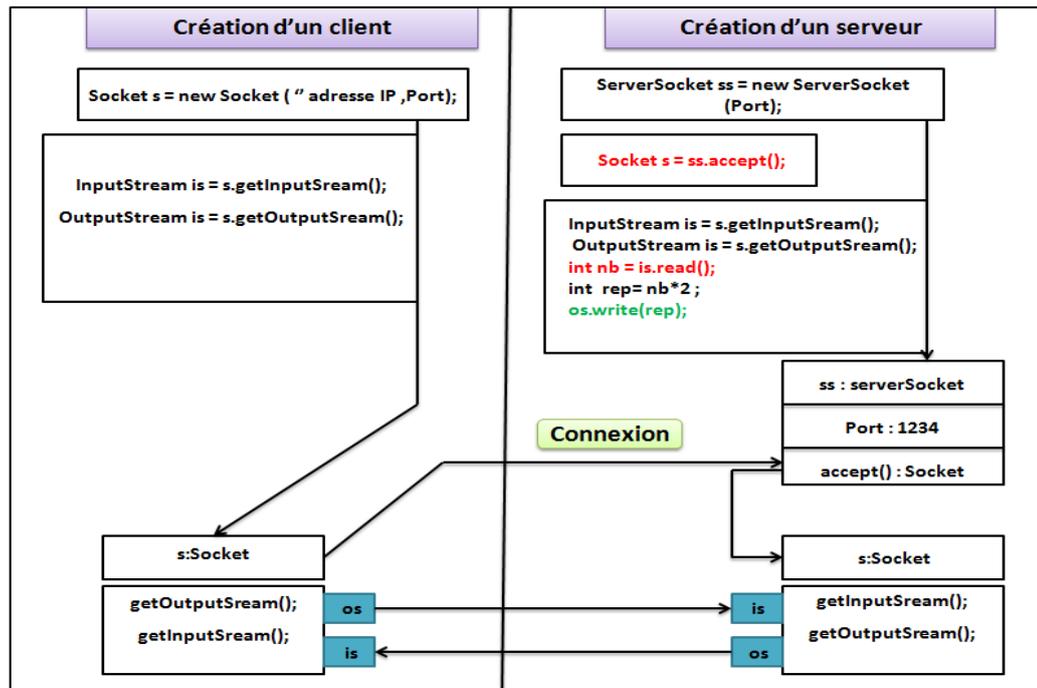


Figure III. 2 : Comment créer un client et serveur simple.

III.3.1. Côté serveur

Soient les étapes suivantes :

Étape 0 : bibliothèques à inclure

Pour utiliser les fonctions de la bibliothèque *Socket*, il est nécessaire d'importer une ou plusieurs bibliothèques de fonctions qui contiennent les primitives que nous utiliserons ci-dessous :

```
Import java.net. Socket;
Import java.io.*;
```

Étape 1 : Créer une socket serveur TCP et définir le numéro de port.

```
ServerSocket serversocket = new ServerSocket (Port);
```

Étape 2 : Permettre à l'application (côté serveur) de prendre connaissance d'une nouvelle connexion.

```
Socket socket = serversocket.accept () ;
```

Au retour de l'appel à *accept*, cette variable contient une référence vers un objet Socket permettant de dialoguer avec le client, aussi la connexion est retirée de la file d'attente.

Étape 3 : Une fois la connexion établie, le client et le serveur peuvent commencer à envoyer et recevoir des flux de données.

```
InputStream is = socket. getInputStream ();  
OutputStream os = socket. getOutputStream ();  
int nb = is. read ();  
os. write (nb);
```

À partir de chaque extrémité, on peut créer deux flux pour interagir avec l'interlocuteur distant :

- Un flux d'entrée (en lecture) : réception/consommation des données émises par l'interlocuteur
- Un flux de sortie (en écriture) : émission/production de données vers l'interlocuteur.

Étape 4 : Pour fermer la connexion en utilise la fonction *close*.

```
socket. close ();
```

III.3.2. Côté client

Soient les étapes ci-après :

Étape 1 : Créer un Socket, et définir l'adresse IP et le numéro de port.

```
Socket socket = new Socket (" address IP", Port);
```

Étape 2 : Créer deux flux : un flux d'entrée (en lecture) pour réception/consommation des données émises, flux de sortie (en écriture) pour émission/production de données.

```
InputStream is = socket. getInputStream ();  
OutputStream is = socket. getOutputStream ();  
os.write(nb);  
int rep = is.Read();
```

Étape 4 : Pour fermer la connexion en utilisant la fonction *close*.

```
serversocket. close ();  
socket. close ();
```

Il faut noter bien que dans les méthodes *InputStream* et *OutputStream* les flux de communication ne dépassent pas 1 octet (8bit).

III.4. Transfert des chaînes de caractère client /serveur

Afin de sarmenter le problème de la limitation de flux de communication, dans cette partie nous allons utiliser des fonctions qui permettent l'envoi des chaînes de caractère.

Etape 1 : Création de l'objet *ServerSocket*

```
ServerSocket ss = new ServerSocket (PORT);
```

Etape 2 : Attendre une connexion d'un client

```
ServerSocket sock =ss.accept();
```

Etape 3 : Pour envoyer une chaîne de caractère au client.

```
OutputStream os = sock.getOutputStream();  
PrintWriter pw = new PrintWriter(os, true);  
pw. println ("chaîne de caractère " );
```

Etape 4 : Pour lire une chaîne de caractère envoyer par le client.

```
InputStream is = sock.getInputStream();  
InputStreamReader isr = sock.getInputStreamReader(is);  
BufferedReader br = new BufferedReader(isr);  
String s =br. readLine();
```

III.5. Transfert des objets entre Client - Serveur (Sérialisation d'objets)

III.5.1. Définition

Sérialiser un objet consiste à le convertir en un tableau d'octets, que l'on peut ensuite écrire dans un fichier, l'envoyer sur un réseau à travers une socket etc... Ce mécanisme existe depuis les débuts de l'API Java I/O, et il est très pratique. Il suffit de passer tout objet qui implémente l'interface *Serializable* à une instance d'*ObjectOutputStream* pour sérialiser un objet. L'opération inverse qui consiste à créer une nouvelle instance à partir du résultat d'une sérialisation s'appelle la désérialisation [14].

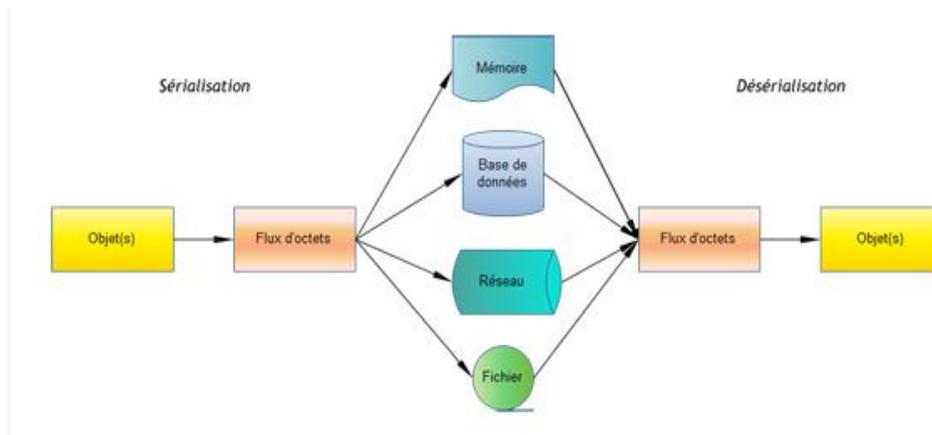


Figure III. 3 : Sérialisation et désérialisation des objets [15].

III.5.2. Objet sérialisable

Soient les étapes suivantes :

Étape 1 : Un objet doit implémenter l'interface *Serializable*

```
Public class Voiture Implement Serializable{
String mat: int carburant;
Public Voiture (string m, int c) {matm; carburant c}
}
```

Étape 2 : La classe *ObjectOutputStream* est une classe de fonctionnalités qui permet d'écrire des objets dans un flux binaire à l'aide de la méthode *writeObject*. Symétriquement la classe *ObjectInputStream* est une classe de fonctionnalités qui permet de lire des objets dans un flux binaire à l'aide de la méthode *readObject*.

```
//envoyer un objet vers le client
OutputStream os = sock.getOutputStream( );
ObjectOutputStream oos = new ObjectOutputStream(os);
Voiture v1 = new Voiture ("v212,50 ' ');
oos. writeObject (v1) ;
// Pour lire un objet envoyé par le client
InputStream is =sock. getInputStream ( );
ObjectInputStream ois = new ObjectInputStream(is);
Voiture v = (Voiture) ois. readObject ( );
```

III.6. Socket Multithread

III.6.1. Définition

Un serveur ayant plusieurs threads est appelé serveur multithread. Lorsqu'un client envoie la demande, un fil est généré à travers lequel un utilisateur peut communiquer avec le serveur. Nous devons générer plusieurs threads pour accepter plusieurs demandes de plusieurs clients en même temps [16].

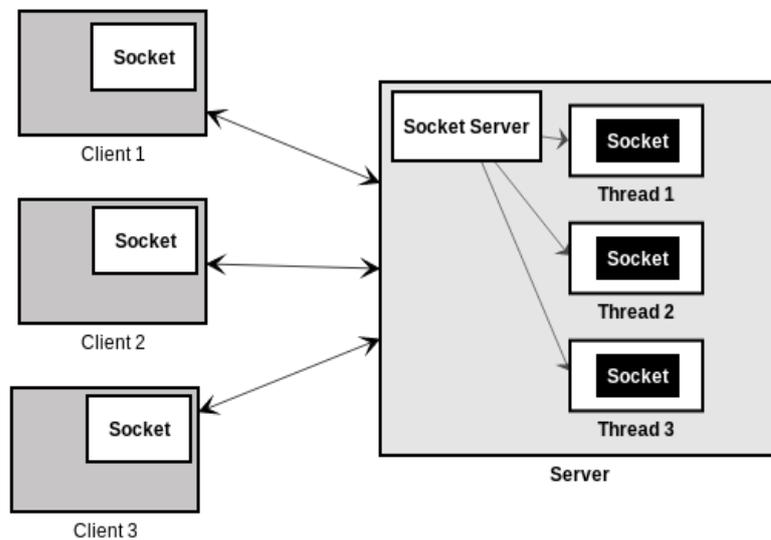


Figure III. 4 : Communication multithread [17].

III.6.2. Avantages du serveur multithread

- **Rapide et efficace** : le serveur multithread pourrait répondre rapidement et efficacement aux requêtes croissantes des clients.
- **Le temps d'attente pour les utilisateurs diminue** : dans un serveur monthread, les autres utilisateurs devaient attendre la fin du processus en cours, mais dans les serveurs multithreads, tous les utilisateurs peuvent obtenir une réponse en même temps, de sorte qu'aucun utilisateur n'a à attendre la fin des autres processus.
- **Les threads sont indépendants les uns des autres** : il n'y a aucune relation entre deux threads. Lorsqu'un client est connecté, un nouveau thread est généré à chaque fois.
- **Le problème dans un thread n'affecte pas les autres threads** : si une erreur se produit dans l'un des threads, aucun autre thread n'est perturbé, tous les autres processus continuent de s'exécuter normalement. Dans un serveur à thread unique, tous les autres clients devaient attendre si un problème survenait dans le thread [16].

III.6.3. Comment déclarer un thread

Pour utiliser Thread il faut deux étapes de base doivent être suivies :

Étapes 1 : Les objets doivent hériter de la classe Thread.

```
Public classe ServeurMT extends Thread
```

Étapes 2 : Déclarez la méthode run(). Le programme à exécuter doit être à l'intérieur de cette méthode.

```
@Override  
Public void run() ; {  
}
```

Étapes 3 : Si vous voulez démarrer votre programme, appelez simplement la méthode start().

```
new serveurMT().start() ;
```

III.7. Application de gestion d'administration

Il s'agit d'une application programmée en Java et reliée à une base de données SQL. L'application permet de gérer les étudiants dans différentes filières et différents niveaux.

Ce travail consiste à développer une application d'interaction qui distribue des fonctionnalités de gestion des étudiants et aussi des enseignants au niveau de la faculté des sciences de notre université. Le but étant d'informatiser notre administration. Et ce, en numérisant complètement le système existant afin de faciliter le contact et la communication entre enseignants, étudiants et administration.

III.7.1. Interfaces Coté Serveur

Dans cette partie nous allons montrer les interfaces homme machine coté serveur.

III.7.1.1. Connexion de l'administrateur (Serveur)

Lors de l'exécution de l'application, l'administrateur va se connecter à l'aide de ses informations d'identification tant qu'administrateur du système.



Figure III. 5 : Connexion du serveur.

Une fois que l'administrateur est connecté, il est redirigé vers la page d'accueil où l'on peut voir de fonctionnalités diverses (Fig. III.6).

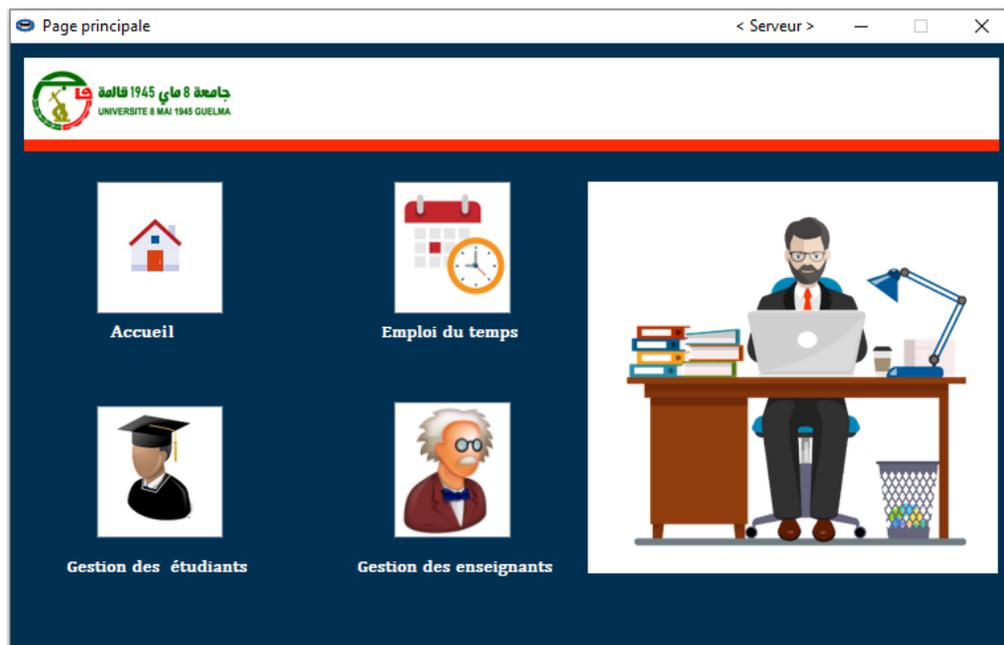


Figure III. 6 : Page principale de l'administrateur.

III.7.1.2. Gestion des étudiants

Dans le module gestion des étudiants l'administrateur peut supprimer, modifier et ajouter de nouveaux étudiants. Si l'administrateur veut imprimer des listes des étudiants, il suffit de cliquer sur l'icône de l'imprimante.

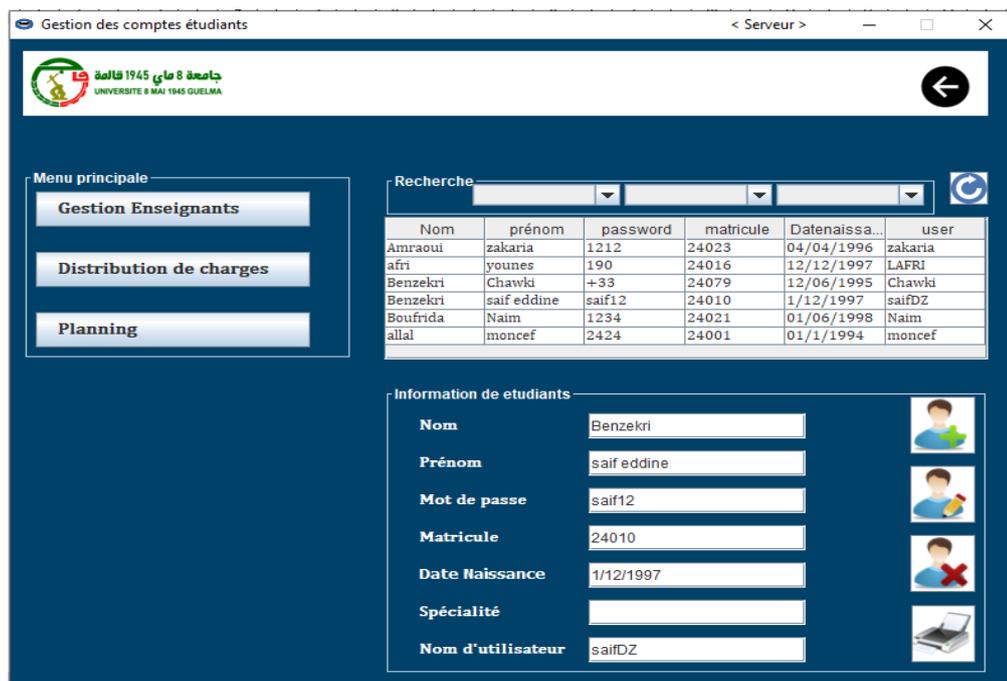


Figure III. 7 : Gestion des comptes étudiants.

III.7.1.3. Gestion des enseignants

En termes de gestion des enseignants, l'administrateur a la possibilité de supprimer et modifier et ajouter de nouveaux enseignants.

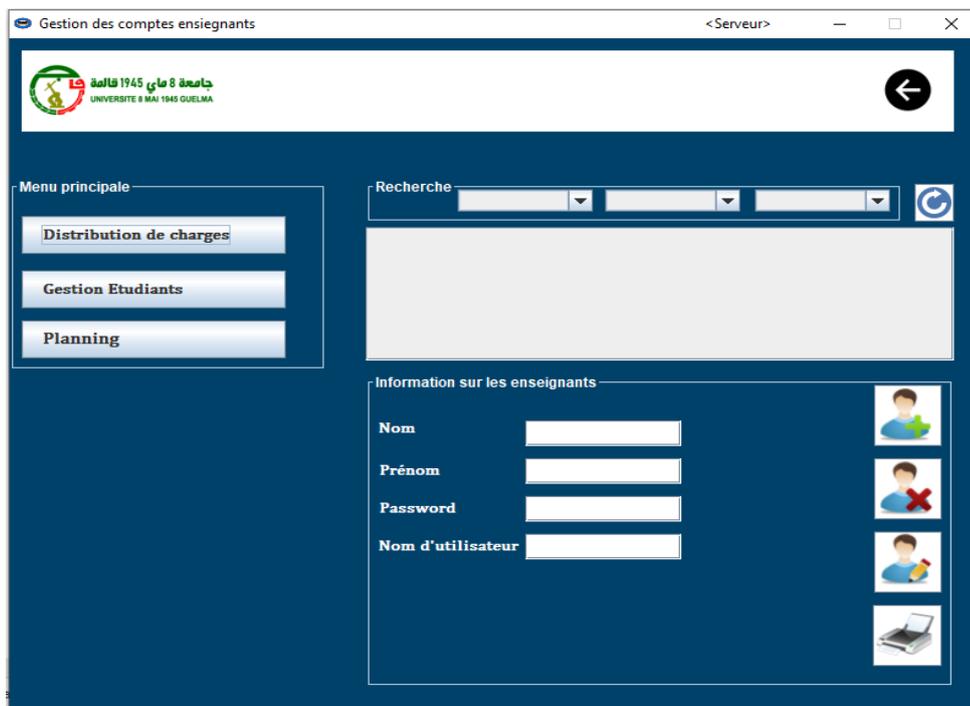


Figure III. 8 : Gestion des comptes enseignants.

Si l'utilisateur veut imprimer des listes des enseignants, il suffit de cliquer sur l'icône de l'imprimante.

III.7.1.4. Gestion de la distribution de charges

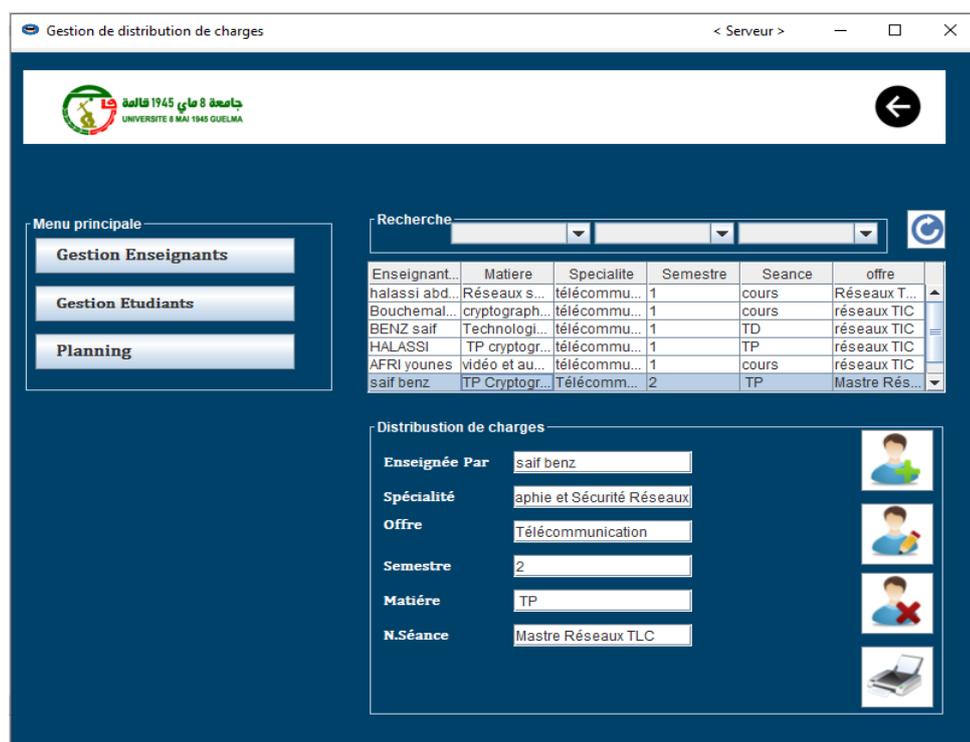


Figure III. 9 : Gestion distribution de charges.

III.7.1.5. Emploi du temps

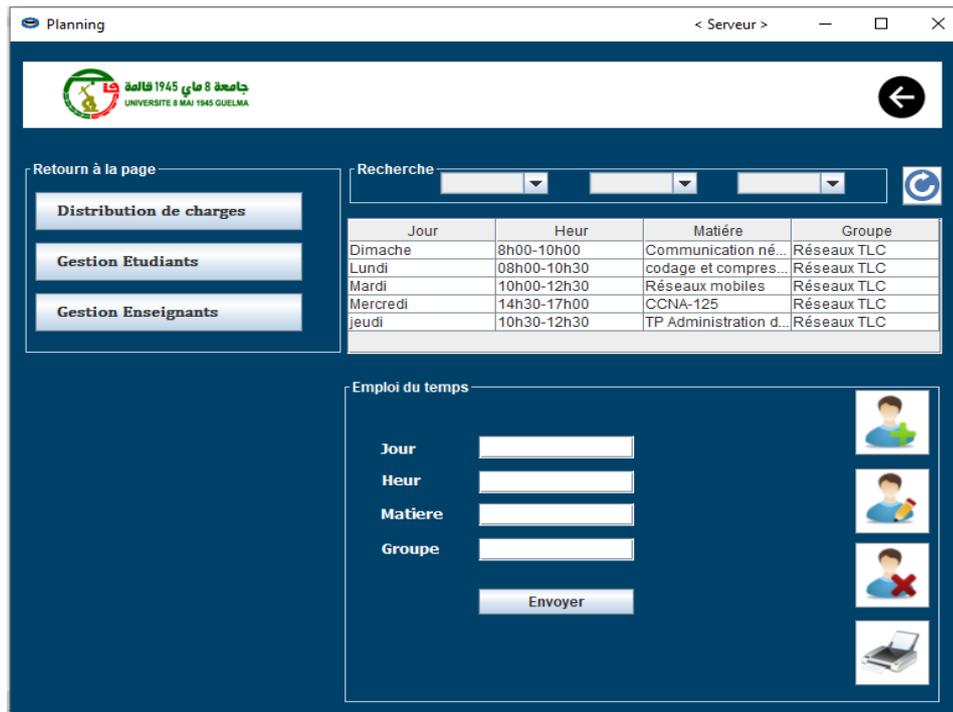


Figure III. 10 : L'emploi du temps.

III.7.2. Interface coté clients

Les interfaces coté clients sont comme suit :

III.7.2.1. Connexion de client

Lors de l'exécution de l'application client, le système invite l'utilisateur à se connecter en saisissant l'adresse IP du serveur dans le champ vide et en sélectionnant son type de compte (compte étudiant ou compte enseignant).

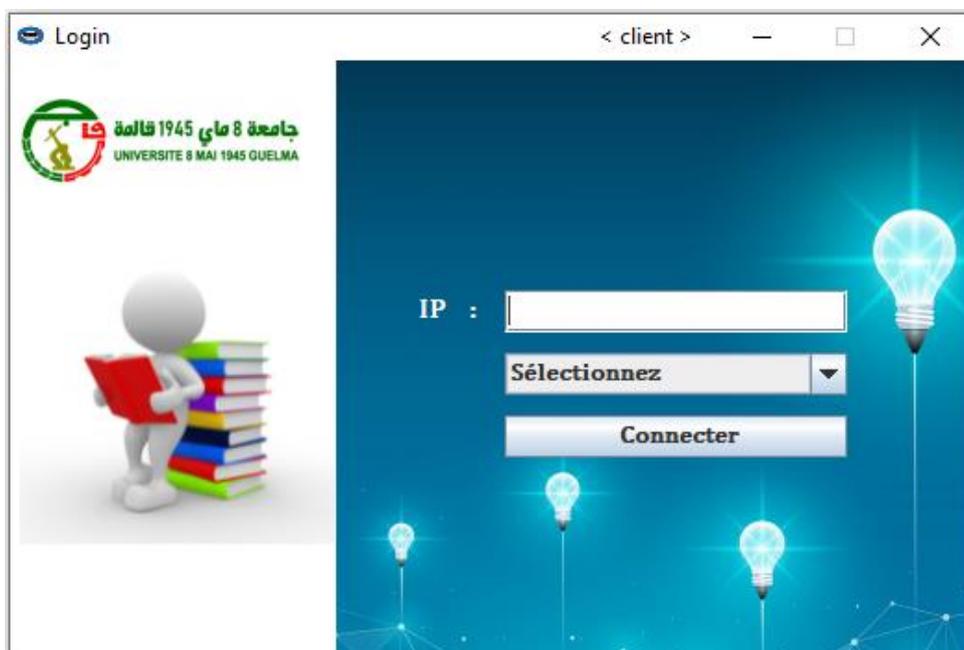


Figure III. 11 : Création un compte étudiant.

III.7.2.2. Création d'un compte étudiant et compte enseignant

Lorsque vous choisissez un type de compte, une interface s'ouvre pour vous en fonction du type de compte que vous avez sélectionné.



Figure III. 12 : Création un compte étudiant.

Si c'est la première fois que vous entrez ici, il vous suffit de vous inscrire en remplissant les champs vides pour créer un compte étudiant (de la même façon pour un compte enseignant).

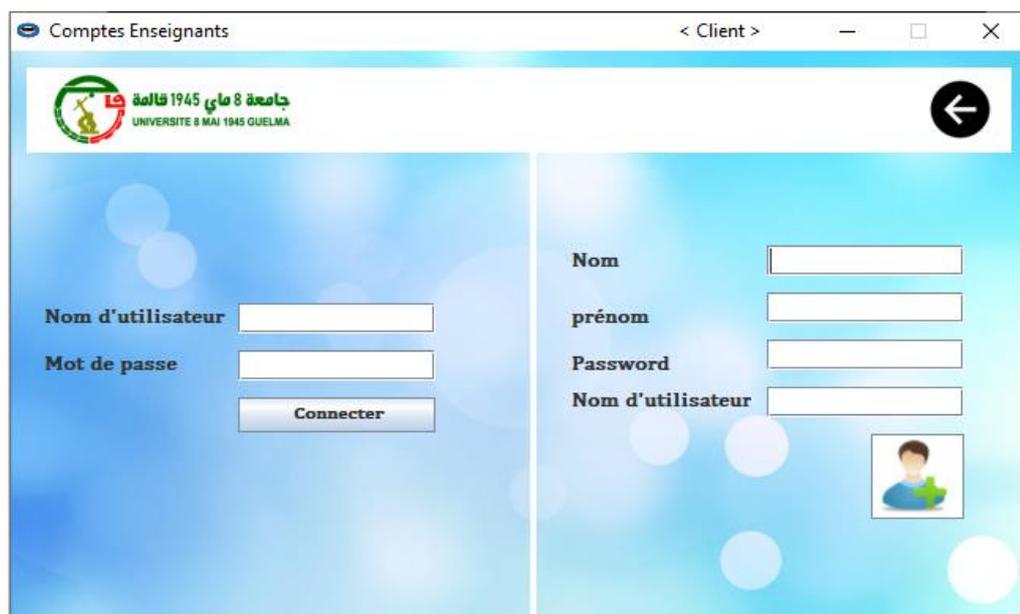


Figure III. 13 : Page pour authentification des enseignants.

Si vous êtes déjà inscrit, il vous suffit de saisir votre nom d'utilisateur et votre mot de passe pour le type de comptes qui vous convient (comptes étudiants ou comptes enseignants).

III.7.2.3. Comptes enseignants

Lorsque vous cliquez sur Se Connecter, une nouvelle interface représentant l'enseignant apparaît, il peut :

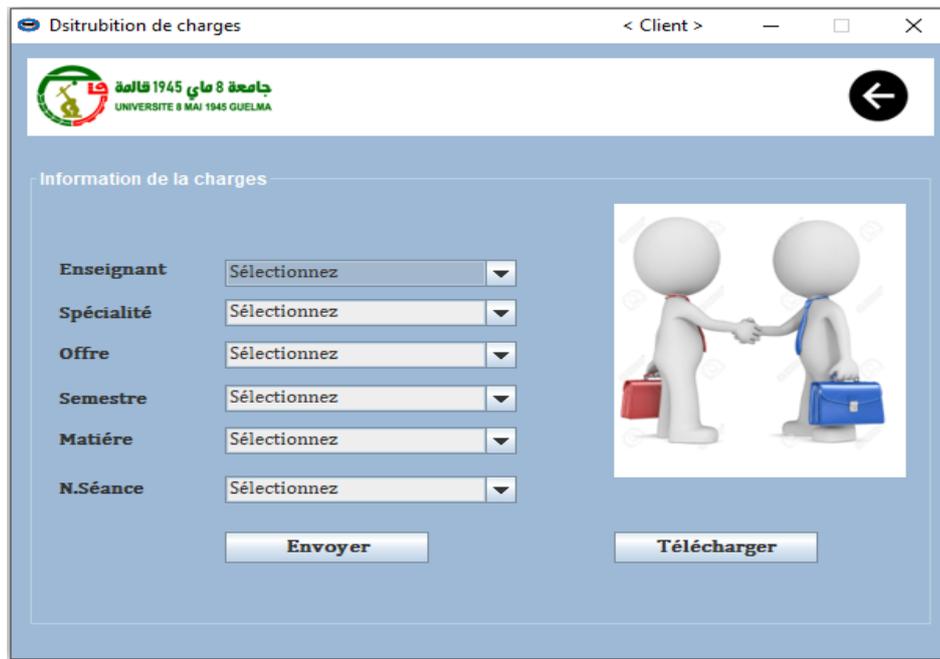
- Voir l'emploi du temps.
- Sélectionner le type de matière à enseigner (dans l'interface de répartition des charges).



Figure III. 14 : Page principale pour les enseignants.

III.7.2.4. Distribution de charges

Lors du choix de l'icône distribution de charges, une interface de distribution apparaît, grâce à laquelle les enseignants peuvent choisir les matières et spécialité qu'ils souhaitent enseigner, ainsi que la nature de séance (cours, TD ou TP).



The screenshot shows a web browser window titled "Dsitribution de charges" with a "< Client >" button. The header features the logo of "جامعة 8 ماي 1945 قالمة" (UNIVERSITE 8 MAI 1945 GUELMA) and a back arrow. The main content area is titled "Information de la charges" and contains a form with the following fields:

- Enseignant: Sélectionnez
- Spécialité: Sélectionnez
- Offre: Sélectionnez
- Semestre: Sélectionnez
- Matière: Sélectionnez
- N.Séance: Sélectionnez

Below the form are two buttons: "Envoyer" and "Télécharger". To the right of the form is an illustration of two 3D figures shaking hands, one with a red briefcase and the other with a blue briefcase.

Figure III. 15 : Distribution de charges.

III.7.2.5. Comptes étudiants

Lorsque vous cliquez sur Se Connecter, une nouvelle interface représentant le compte de l'étudiant il peut vu des notes d'examen ainsi que l'emploi du temps.

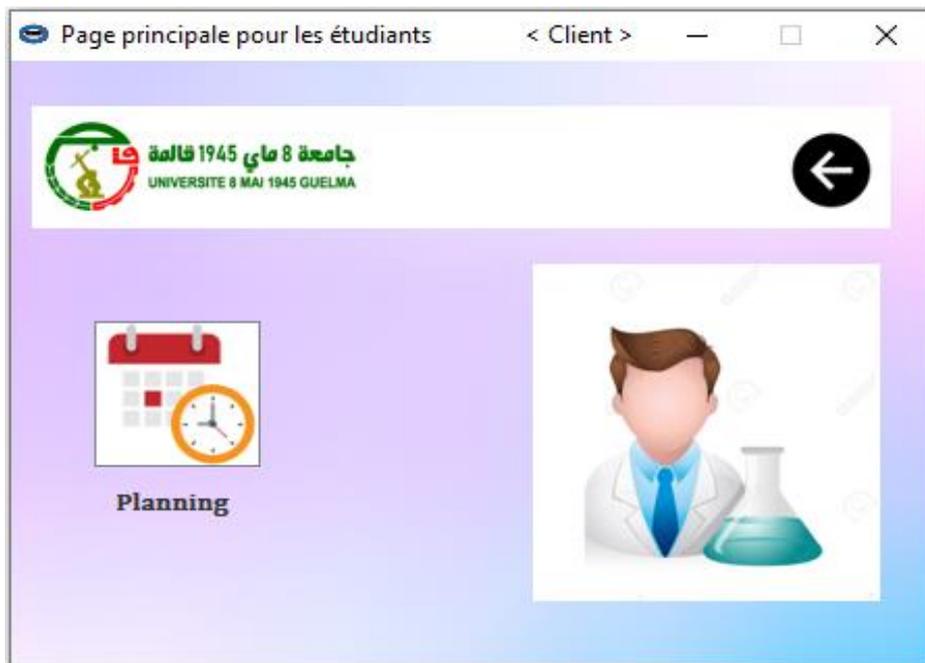


Figure III. 16 : Page principale pour les étudiants.

III. 8. Conclusion

Dans ce dernier chapitre, nous l'avons subdivisé en deux parties, la première partie a concerné la programmation en utilisant le langage JAVA, en particulier nous avons commencé par apprendre comment établir une communication entre Client et Serveur en créant des applications via Sockets.

La deuxième partie a été dédiée en la présentation de notre application de gestion d'administration, qui offre trois interfaces de gestion des étudiants, des enseignants et distribution des charges, à travers ces interfaces il peut y avoir des interactions entre étudiants et enseignants notamment les changements de planning, consultation des notes ...etc.

CONCLUSION GÉNÉRALE

Ce travail a été le fruit de nos efforts consentis et des encouragements de nos cadres enseignants pour la réalisation de ce petit document faisant un mémoire de fin d'études dans le domaine des réseaux informatique. En effet, il nous a permis d'une part de comprendre les notions de bases de ses réseaux de télécommunication, d'autre part d'acquérir des nouvelles connaissances notamment dans la programmation via le langage Java.

En fait, les technologies de télécommunications utilisées dans les réseaux locaux ont connu un large succès qui est notamment dû à la facilité de configuration et aux prix avantageux. Ce succès a encouragé l'adaptation de ces technologies à des contextes différents tels que les réseaux métropolitains ou publics.

A travers les chapitres de ce mémoire, on a essayé de toucher l'essentiel sur les notions fondamentales des réseaux informatiques les plus connus, à savoir leurs architectures, les protocoles utilisés et comment fonctionnent les types de réseaux existents (LAN, WAN, VPN), en suite on s'est intéressé essentiellement au modèle de base de tous les réseaux informatiques qui est le modèle client-serveur, en fait, on a expliqué comment créer un réseau local LAN grâce à l'utilisation des sockets. D'un point de vue pratique, on a beaucoup apprécié notre évolution au codage en Java, qui s'est traduit en la réalisation d'une application de gestion administrative pour les étudiants et les enseignants, qui leurs offre un choix de communication intéressant, facile et rapide.

Par ailleurs, d'un point de vu global ce projet nos a bien apprendre la gestion d'un projet, la réflexion sur la manière de le réaliser et à quel point faut-il donner de l'impotence aux compétences acquises ainsi que aux orientations données pour se préparer à la vie professionnel.

Enfin, nous espérons que ce modeste travail constitue une initiation à tous ceux qui voudront s'engager dans son amélioration.

REFERENCES BIBLIOGRAPHIQUES

- [1] « Un aperçu des différents réseaux informatiques. », IONOS Digital guide, <https://www.ionos.fr/digitalguide/serveur/know-how/les-types-de-reseaux-informatiques-a-connaître>, consulté le : 18 avr. 2021.
- [2] « Topologie des réseaux - Comment Ça Marche », <https://www.commentcamarche.net/contents/512-topologie-des-reseaux> , consulté le : 18avr. 2021.
- [3] Site internet « www.STI2D.com »
- [4] François Laissus « fr .laissus [at] laissus . fr » , Version du 20 février 2005
- [5] S. BELKHOUCHE, « Etude et Administration des Systèmes de Supervision dans un Réseau Local », Université Abou Bakr Belkaid– Tlemcen, Tlemcen-Algérie, 2010.
- [6] « Réseau Local filaire basés sur technologie Ethernet» <https://www.wixxim.fr/fiches/le-reseau-local-lan> , consulté le : 19 juin 2021.
- [7] « Réseau LAN Schéma Illustrator pour les affaires et la technologie Concept », 123RF. https://fr.123rf.com/photo_23981367_r%C3%A9seau-lan-sch%C3%A9ma-illustrator-pour-les-affaires-et-la-technologie-concept.html, consulté le : 19 juin 2021.
- [8] « Socket : Communication via le réseau – Recherche Google », consulter le : 02 juillet 2021. https://www.u-picardie.fr/ferment/java/chap32_c.html, consulté le 19 juin 2021.
- [9] Page web « <https://www.ibm.com/docs/fr/rtw/9.2.1?topic=transports-sockets-overview> » consulté le : 12 juin 2021.
- [10] Page web : <https://arabicprogrammer.com/article/31141783183>, consulté le : 04 juillet 2021.
- [11] « Service MySQL Database | Oracle Algeria » <https://www.oracle.com/dz/mysql/> consulté le : 04 juillet 2021.
- [12] Page web « <https://web.maths.unsw.edu.au/~lafaye/CCM/bdd/bddintro.htm>», consulté le : 04 juillet 2021.
- [13] Page web « <https://sql.sh/> « Cours et Tutoriels sur le Langage SQL », consulté le : 29 juin 2021.
- [14] « Serialization d'objets » <http://blog.paumard.org/cours/java/chap10-entrees-sorties-serialization.html>, consulté le : 04 juillet 2021.
- [15] « Séri­alisation et déséri­alisation des objets. », <https://www.jmdoudoux.fr/java/dej/chap-serialisation.htm>, consulté le : 25 juin 2021.
- [16] « Multithread Servers in Java. », <https://www.geeksforgeeks.org/multithreaded-servers-in-java>, consulté le : 25 Juin 2021.
- [17] « [java-socket-programming-process.jpg\(700×383\)](https://www.geeksforgeeks.org/multithreaded-servers-in-java) », [java-socket-programming-process.jpg](https://www.geeksforgeeks.org/multithreaded-servers-in-java) , consulté le: 25 Avril 2021.

