



# Projet Réseaux

LAMOOT David

## Table of Contents

<b>Introduction .....</b>	<b>3</b>
<b>TCP.....</b>	<b>3</b>
<b>Description .....</b>	<b>3</b>
Serveur .....	3
Client .....	4
<b>Test .....</b>	<b>5</b>
<b>UDP.....</b>	<b>8</b>
<b>Description .....</b>	<b>8</b>
Serveur .....	8
Client .....	8
<b>Test .....</b>	<b>9</b>
<b>Conclusion .....</b>	<b>12</b>

## Introduction

Dans le cadre de ce projet, nous avons développé une application de gestion de comptes bancaires reposant sur une architecture client-serveur. Ce projet avait pour objectif principal d'implémenter et d'évaluer les fonctionnalités de gestion de comptes bancaires à l'aide de sockets TCP et UDP sous Linux. Les fonctionnalités incluent la consultation des soldes, la gestion des transactions (ajout et retrait de fonds) et l'historique des opérations, tout en garantissant une interaction sécurisée entre le client et le serveur. Ce rapport détaille la conception, l'implémentation, et les résultats des tests réalisés.

## TCP

### Description

#### Serveur

- 1) Initialisation du Socket : Le code commence par créer un socket TCP en utilisant la fonction `socket()`. Ce socket est configuré avec les paramètres réseau appropriés, notamment l'adresse IP et le numéro de port. La fonction `bind()` associe ensuite cette socket à l'adresse et au port spécifiés, permettant au serveur de recevoir des paquets sur cette adresse.
- 2) Authentification : Le serveur reçoit l'identifiant, le numéro de compte et le mot de passe du client via le socket TCP. Ces informations sont comparées à des données d'authentification préenregistrées. Si les identifiants concordent, l'accès est accordé. Cette étape est cruciale pour la sécurité, bien que dans un contexte réel, une méthode plus sécurisée serait employée.
- 3) Boucle Principale : Le serveur entre dans une boucle infinie, attendant les requêtes des clients. Il traite différentes requêtes telles que la connexion, les transactions de crédit ou de débit, et la déconnexion. Cette boucle représente le cœur du serveur, traitant activement les interactions client-serveur.

4) Gestion des Transactions : Lorsqu'une requête de transaction est reçue, le serveur effectue l'opération demandée (crédit ou débit) sur le solde du compte. Il tient également à jour un historique des transactions, ce qui permet de suivre les mouvements de fonds sur chaque compte.

5) Réponses aux Clients : Après chaque opération, le serveur envoie une réponse au client. Cette réponse peut inclure la confirmation d'une transaction, le solde actuel du compte, ou un historique des transactions récentes. Ces informations sont transmises via le socket TCP au client.

#### Client

1) Établissement de la Connexion : Le client initialise une connexion TCP avec le serveur. Il configure son socket avec les paramètres du serveur (adresse IP et port) et utilise ce socket pour communiquer avec le serveur. Cette étape est essentielle pour établir le chemin de communication entre le client et le serveur.

2) Interface Utilisateur : L'application client offre une interface simple en ligne de commande. L'utilisateur est invité à entrer son identifiant et son mot de passe, puis à sélectionner une option parmi celles proposées dans le menu. Cette interface utilisateur assure l'interaction avec l'application serveur.

3) Envoi des Requêtes : Basé sur le choix de l'utilisateur, le client envoie des requêtes au serveur. Ces requêtes peuvent être des demandes de transaction, des consultations de solde, ou d'autres opérations bancaires. Le client utilise la même socket TCP pour envoyer ces requêtes.

4) Réception et Affichage des Réponses : Le client reçoit les réponses du serveur via le socket TCP. Ces réponses incluent des confirmations de transactions réussies, le solde actuel du compte, et l'historique des transactions. Le client affiche ces informations à l'utilisateur.

5) Logique Implémentée : Ces descriptions reflètent la logique fondamentale du fonctionnement du client et du serveur, tel que mis en œuvre dans `serveur.c` et `client.c`. La communication et la coordination entre le client et le serveur sont au cœur de cette application.

## Test

Voici les différents tests effectués allant d'erreur volontaire à l'utilisation complète de l'application.

```

davidlamoot — client — 105x31
Last login: Sun Jan 12 22:26:36 on ttys001
/Users/davidlamoot/.zprofile:1: no such file or directory: /opt/homebrew/bin/brew
/Users/davidlamoot/Library/CloudStorage/OneDrive-etu.sorbonne-universite.fr/Cours/EISE4/S7/Réseaux/Projet
\reseaux/Projet\reseaux/TCP/client ; exit;
(base) davidlamoot@pc-507 ~ % /Users/davidlamoot/Library/CloudStorage/OneDrive-etu.sorbonne-universite.fr
/Cours/EISE4/S7/Réseaux/Projet\reseaux/Projet\reseaux/TCP/client ; exit;
[+] Socket client cree et connexion au serveur avec succes
=====
-Saisissez votre identifiant :
test
=====
-Saisissez votre numero de compte :
hein
=====
-Saisissez votre mot de passe :
84156
=====

Mot de passe ou identifiant incorrecte

Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[Process completed]
```

```

davidlamoot — client — 105x31
Last login: Sun Jan 12 22:26:36 on ttys001
/Users/davidlamoot/.zprofile:1: no such file or directory: /opt/homebrew/bin/brew
/Users/davidlamoot/Library/CloudStorage/OneDrive-etu.sorbonne-universite.fr/Cours/EISE4/S7/Réseaux/Projet
\reseaux/Projet\reseaux/TCP/client ; exit;
(base) davidlamoot@pc-507 ~ % /Users/davidlamoot/Library/CloudStorage/OneDrive-etu.sorbonne-universite.fr
/Cours/EISE4/S7/Réseaux/Projet\reseaux/Projet\reseaux/TCP/client ; exit;
[+] Socket client cree et connexion au serveur avec succes
=====
-Saisissez votre identifiant :
test
=====
-Saisissez votre numero de compte :
hein
=====
-Saisissez votre mot de passe :
84156
=====

Mot de passe ou identifiant incorrecte

Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[Process completed]
```

```
davidlamoot — serveur — 111x36

/Users/davidlamoot/.zprofile:1: no such file or directory: /opt/homebrew/bin/brew
/Users/davidlamoot/Library/CloudStorage/OneDrive-etu.sorbonne-universite.fr/Cours/EISE4/S7/Réseaux/Projet\!reseaux/Projet\!reseaux/TCP/serveur ; exit;
(base) davidlamoot@pc-507 ~ % /Users/davidlamoot/Library/CloudStorage/OneDrive-etu.sorbonne-universite.fr/Cours/EISE4/S7/Réseaux/Projet\!reseaux/Projet\!reseaux/TCP/serveur ; exit;
[+]Socket serveur cree avec succes
[+]bind : 3
[+]listen...
[+]On accepte
[+]client: 4
*****
-Saisissez votre identifiant :
vous etes bien : test
*****
-Saisissez votre numero de compte :
0000001
*****
-Saisissez votre mot de passe :
hein
*****
-Bienvenue sur votre compte bancaire

1 Le client ajoute 25
2 Le client retire 2000
3 Le client consulte sa solde
4 Le client consulte l'historique
FIN
close

Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[Process completed]
```

```
davidlamoot — client — 104x31

Appuyez sur 1 si vous souhaitez ajouter de l'argent dans votre compte
Appuyez sur 2 si vous souhaitez retirer de l'argent dans votre compte
Appuyez sur 3 si vous souhaitez consulter votre solde
Appuyez sur 4 si vous souhaitez consulter les 10 dernieres operations effectuees sur votre compte
Appuyez sur 5 si vous souhaitez quitter l'interface
4

Voici vos 10 dernieres operations sur votre compte
affichage solde
retrait 146 €
ajout 2146 €

-----

Appuyez sur 1 si vous souhaitez ajouter de l'argent dans votre compte
Appuyez sur 2 si vous souhaitez retirer de l'argent dans votre compte
Appuyez sur 3 si vous souhaitez consulter votre solde
Appuyez sur 4 si vous souhaitez consulter les 10 dernieres operations effectuees sur votre compte
Appuyez sur 5 si vous souhaitez quitter l'interface
5
-----FIN D'OPERATION BONNE JOURNEE-----
```

```

davidlamoot — client — 104x31
Appuyez sur 1 si vous souhaitez ajouter de l'argent dans votre compte
Appuyez sur 2 si vous souhaitez retirer de l'argent dans votre compte
Appuyez sur 3 si vous souhaitez consulter votre solde
Appuyez sur 4 si vous souhaitez consulter les 10 dernieres operations effectuees sur votre compte
Appuyez sur 5 si vous souhaitez quitter l'interface
2
*****
Solde sur votre compte : 2146 €
Veuillez saisir la somme a retirer :
2000
*****
Votre solde est desormais de 146 € sur votre compte

-----FIN D'OPERATION-----

Appuyez sur 1 si vous souhaitez ajouter de l'argent dans votre compte
Appuyez sur 2 si vous souhaitez retirer de l'argent dans votre compte
Appuyez sur 3 si vous souhaitez consulter votre solde
Appuyez sur 4 si vous souhaitez consulter les 10 dernieres operations effectuees sur votre compte
Appuyez sur 5 si vous souhaitez quitter l'interface
3
*****
Solde sur votre compte : 146 €

-----FIN D'OPERATION-----

```

```

davidlamoot — client — 109x35
[+] Socket client cree et connexion au serveur avec succes
-----
-Saisissez votre identifiant :
test
-----
-Saisissez votre numero de compte :
00000001
-----
-Saisissez votre mot de passe :
hein
-----
-Bienvenue sur votre compte bancaire

-----

Appuyez sur 1 si vous souhaitez ajouter de l'argent dans votre compte
Appuyez sur 2 si vous souhaitez retirer de l'argent dans votre compte
Appuyez sur 3 si vous souhaitez consulter votre solde
Appuyez sur 4 si vous souhaitez consulter les 10 dernieres operations effectuees sur votre compte
Appuyez sur 5 si vous souhaitez quitter l'interface
1
-----
Solde sur votre compte : 2121 €
Veuillez saisir la somme ajoute :
25
*****
Votre solde est desormais de 2146 € sur votre compte

-----FIN D'OPERATION-----

```

## UDP

### Description

#### Serveur

1) Initialisation du Socket : Le code commence par créer un socket UDP en utilisant la fonction `socket ()`. Ce socket est configuré avec les paramètres réseau appropriés, notamment l'adresse IP et le numéro de port. La fonction `bind ()` associe ensuite cette socket à l'adresse et au port spécifiés, permettant au serveur de recevoir des paquets sur cette adresse.

2) Authentification : Le serveur reçoit l'identifiant et le mot de passe du client via la socket UDP. Ces informations sont comparées à des données d'authentification préenregistrées. Si les identifiants concordent, l'accès est accordé. Cette étape est cruciale pour la sécurité, bien que dans un contexte réel, une méthode plus sécurisée serait employée.

3) Boucle Principale : Le serveur entre dans une boucle infinie, attendant les requêtes des clients. Il traite différentes requêtes telles que la connexion, les transactions de crédit ou de débit, et la déconnexion. Cette boucle représente le cœur du serveur, traitant activement les interactions client-serveur.

4) Gestion des Transactions : Lorsqu'une requête de transaction est reçue, le serveur effectue l'opération demandée (crédit ou débit) sur le solde du compte. Il tient également à jour un historique des transactions, ce qui permet de suivre les mouvements de fonds sur chaque compte.

5) Réponses aux Clients : Après chaque opération, le serveur envoie une réponse au client. Cette réponse peut inclure la confirmation d'une transaction, le solde actuel du compte, ou un historique des transactions récentes. Ces informations sont transmises via la socket UDP au client.

#### Client

1) Établissement de la Connexion : Le client initialise une connexion UDP avec le serveur. Il configure son socket avec les paramètres du serveur (adresse IP et port) et utilise ce socket pour communiquer avec le serveur. Cette étape est essentielle pour établir le chemin de communication entre le client et le serveur.



2) Interface Utilisateur : L'application client offre une interface simple en ligne de commande. L'utilisateur est invité à entrer son identifiant et son mot de passe, puis à sélectionner une option parmi celles proposées dans le menu. Cette interface utilisateur assure l'interaction avec l'application serveur.

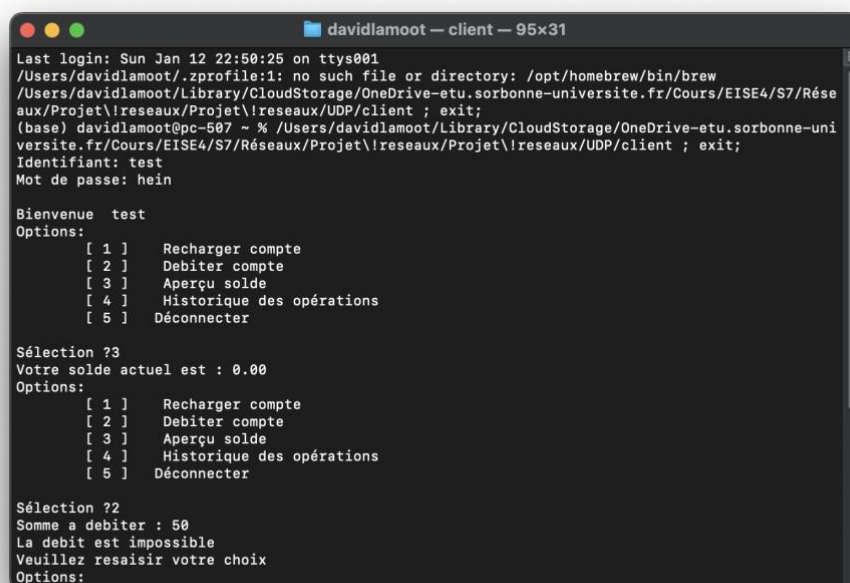
3) Envoi des Requêtes : Basé sur le choix de l'utilisateur, le client envoie des requêtes au serveur. Ces requêtes peuvent être des demandes de transaction, des consultations de solde, ou d'autres opérations bancaires. Le client utilise la même socket UDP pour envoyer ces requêtes.

4 ) Réception et Affichage des Réponses : Le client reçoit les réponses du serveur via le socket UDP. Ces réponses incluent des confirmations de transactions réussies, le solde actuel du compte, et l'historique des transactions. Le client affiche ces informations à l'utilisateur.

5) Logique Implémentée : Ces descriptions reflètent la logique fondamentale du fonctionnement du client et du serveur, tel que mis en œuvre dans serveur.c et client.c. La communication et la coordination entre le client et le serveur sont au cœur de cette application.

## Test

Voici les différents tests effectués allant d'erreur volontaire à l'utilisation complète de l'application.



```
davidlamoot - client - 95x31
Last login: Sun Jan 12 22:50:25 on ttys001
/Users/davidlamoot/.zprofile:1: no such file or directory: /opt/homebrew/bin/brew
/Users/davidlamoot/Library/CloudStorage/OneDrive-etu.sorbonne-universite.fr/Cours/EISE4/S7/Réseaux/Projet\!reseaux/Projet\!reseaux/UDP/client ; exit;
(base) davidlamoot@pc-507 ~ % /Users/davidlamoot/Library/CloudStorage/OneDrive-etu.sorbonne-universite.fr/Cours/EISE4/S7/Réseaux/Projet\!reseaux/Projet\!reseaux/UDP/client ; exit;
Identifiant: test
Mot de passe: hein

Bienvenue test
Options:
[ 1 ] Recharger compte
[ 2 ] Debiter compte
[ 3 ] Aperçu solde
[ 4 ] Historique des opérations
[ 5 ] Déconnecter

Sélection 73
Votre solde actuel est : 0.00
Options:
[ 1 ] Recharger compte
[ 2 ] Debiter compte
[ 3 ] Aperçu solde
[ 4 ] Historique des opérations
[ 5 ] Déconnecter

Sélection 72
Somme a debiter : 50
La debit est impossible
Veuillez resaisir votre choix
Options:
```

```

w
/Users/davidlamoot/Library/CloudStorage/OneDrive-etu.sorbonne-universite.fr/Cours/EISE4/S7/Réseaux/Projet\!reseaux/Projet\!reseaux/UDP/serveur ; exit;
(base) davidlamoot@pc-507 ~ % /Users/davidlamoot/Library/CloudStorage/OneDrive-etu.sorbonne-universite.fr/Cours/EISE4/S7/Réseaux/Projet\!reseaux/Projet\!reseaux/UDP/serveur ; exit;
CONNEXION...
Un user non identifie a essaye de se connecter
L'user test vient de se connecter
L'user test souhaite crediter de 100.000000
L'user test vient de crediter la somme de 100.000000
L'user test souhaite retrait de 50.000000
L'user test vient de debiter la somme de 50.000000
solde : Votre solde actuel est : 50.00
L'user test consulte son SOLDE
L'user test vient de se deconnecter

Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[Process completed]
```

```

Transaction 5 : 0.00
Transaction 6 : 0.00
Transaction 7 : 0.00
Transaction 8 : 0.00
Transaction 9 : 0.00
Transaction 10 : 0.00
Fin de l'historique
Options:
    [ 1 ]   Recharger compte
    [ 2 ]   Debiter compte
    [ 3 ]   Aperçu solde
    [ 4 ]   Historique des opérations
    [ 5 ]   Déconnecter

Sélection ?5
A bientôt test!

Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[Process completed]
```

```
davidlamoot — client — 80x24

Options:
[ 1 ] Recharger compte
[ 2 ] Debiter compte
[ 3 ] Aperçu solde
[ 4 ] Historique des opérations
[ 5 ] Déconnecter

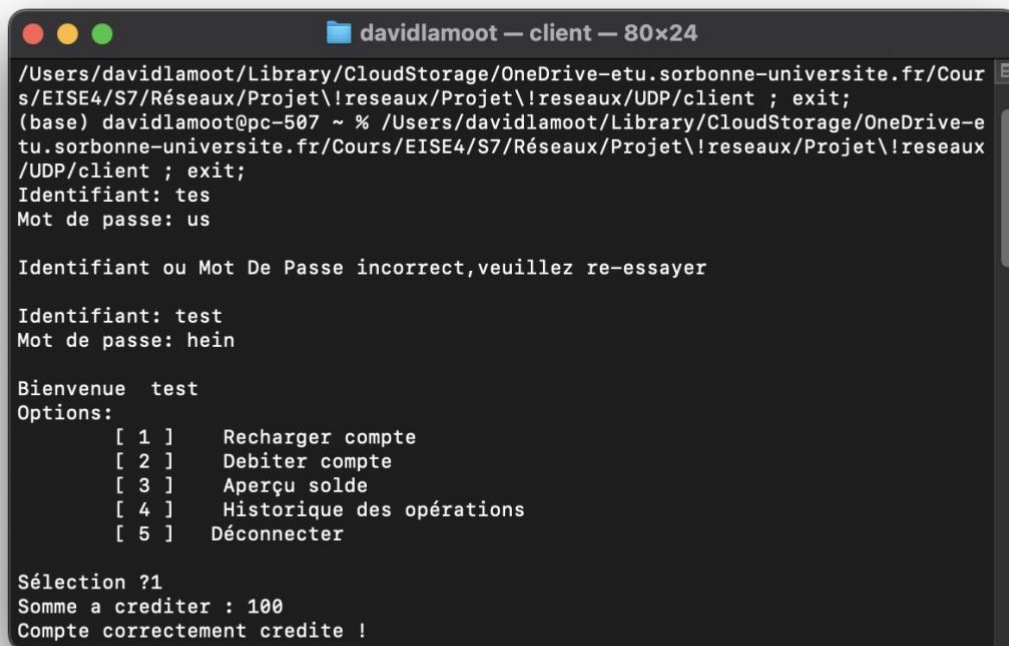
Sélection ?4
Historique des transactions :
Transaction 1 : -50.00
Transaction 2 : 100.00
Transaction 3 : 0.00
Transaction 4 : 0.00
Transaction 5 : 0.00
Transaction 6 : 0.00
Transaction 7 : 0.00
Transaction 8 : 0.00
Transaction 9 : 0.00
Transaction 10 : 0.00
Fin de l'historique
Options:
[ 1 ] Recharger compte
[ 2 ] Debiter compte
[ 3 ] Aperçu solde
```

```
davidlamoot — client — 80x24

Compte correctement credite !
Options:
[ 1 ] Recharger compte
[ 2 ] Debiter compte
[ 3 ] Aperçu solde
[ 4 ] Historique des opérations
[ 5 ] Déconnecter

Sélection ?2
Somme a debiter : 50
Compte correctement debite !
Options:
[ 1 ] Recharger compte
[ 2 ] Debiter compte
[ 3 ] Aperçu solde
[ 4 ] Historique des opérations
[ 5 ] Déconnecter

Sélection ?3
Votre solde actuel est : 50.00
Options:
[ 1 ] Recharger compte
[ 2 ] Debiter compte
[ 3 ] Aperçu solde
```



```
/Users/davidlamoot/Library/CloudStorage/OneDrive-etu.sorbonne-universite.fr/Cours/EISE4/S7/Réseaux/Projet\!reseaux/Projet\!reseaux/UDP/client ; exit;
(base) davidlamoot@pc-507 ~ % /Users/davidlamoot/Library/CloudStorage/OneDrive-etu.sorbonne-universite.fr/Cours/EISE4/S7/Réseaux/Projet\!reseaux/Projet\!reseaux/UDP/client ; exit;
Identifiant: tes
Mot de passe: us

Identifiant ou Mot De Passe incorrect,veuillez re-essayer

Identifiant: test
Mot de passe: hein

Bienvenue test
Options:
    [ 1 ] Recharger compte
    [ 2 ] Debiter compte
    [ 3 ] Aperçu solde
    [ 4 ] Historique des opérations
    [ 5 ] Déconnecter

Sélection ?1
Somme a crediter : 100
Compte correctement credite !
```

## Conclusion

Le projet a permis de mettre en pratique les concepts fondamentaux des architectures client-serveur et l'utilisation des sockets en environnement Linux. L'application implémentée répond aux spécifications et a été testée avec succès pour gérer plusieurs opérations bancaires. Les fonctionnalités de base sont opérationnelles, et une extension pour supporter plusieurs clients simultanés a également été envisagée. Ce projet illustre la pertinence des protocoles TCP et UDP dans des scénarios variés, tout en soulignant l'importance d'une gestion rigoureuse des interactions réseau pour assurer la fiabilité des échanges.