

31/12/2025

SAE302 Développer des applications communicantes

Fiche de documentation technique et réponse au cahier des charges

!!! J'ai rencontré un problème lors de l'envoie de la vidéo car celle-ci était trop volumineuse, je vous transmets donc un lien drive qui vous permettra d'y accéder !!!

Lien vers la Vidéo :

https://drive.google.com/drive/folders/1fm3xeYMA1H11pgZt0YD8uczhPxSFBFvD?usp=drive_link

Table des matières

1. Éléments implémentés et non implémentés	2
Éléments implémentés.....	2
Éléments partiellement implémentés ou non implémentés	2
2. Structure du code, modules et protocole	3
Organisation générale du projet.....	3
Master.....	3
Routeur	3
Client	4
3. Protocole de communication	4
4. Algorithme de chiffrement : principe, forces et faiblesses.....	5
Principe envisagé	5
Forces	5
Faiblesses	5
5. Rapport de projet et gestion du projet	5
Organisation du travail.....	6
Gestion du temps.....	6
Bilan personnel....	6
Conclusion	6

1. Éléments implémentés et non implémentés

Éléments implémentés

Dans le cadre de ce projet, j'ai mis en place une architecture complète de type client / routeurs / master, conforme au cahier des charges initial.

J'ai développé :

- un serveur master chargé de centraliser les informations
- des routeurs jouant le rôle de nœuds intermédiaires
- des clients capables d'envoyer et de recevoir des messages

La communication entre ces éléments repose sur des sockets TCP, ce qui m'a permis d'utiliser un véritable protocole réseau.

J'ai également intégré :

- une base de données MariaDB pour stocker les informations des routeurs
- la récupération dynamique des routeurs par le client
- le routage des messages via plusieurs routeurs intermédiaires
- le déploiement du projet sur plusieurs machines, et non uniquement en local
- des interfaces graphiques pour le client, le routeur et le master, afin de faciliter l'utilisation et les tests

Le système permet ainsi d'envoyer un message depuis un client émetteur, de le faire transiter par plusieurs routeurs, puis de le livrer à un client destinataire.

Éléments partiellement implémentés ou non implémentés

Le chiffrement constitue la principale fonctionnalité incomplète du projet.

J'ai tenté de mettre en place un mécanisme de chiffrement basé sur :

- RSA pour l'échange de clés
- un chiffrement symétrique pour le contenu des messages

Cependant, malgré de nombreux tests et corrections, le chiffrement n'est pas pleinement fonctionnel à la fin du projet. J'ai néanmoins conservé les différentes tentatives et fonctions RSA dans le code afin de montrer le travail réalisé et ma démarche de compréhension.

Par ailleurs :

- la suppression automatique d'un routeur lorsqu'il est éteint brutalement n'est pas implémentée

- la gestion avancée des erreurs réseau reste limitée
 - l'authentification forte des clients et routeurs n'a pas été mise en place
-

2. Structure du code, modules et protocole

Organisation générale du projet

J'ai structuré le projet autour de trois scripts principaux :

- master.py
- router.py
- client.py

Chaque script correspond à un rôle précis et communique avec les autres via le réseau.

Master

Le master est le cœur du système.

Il permet :

- de recevoir les déclarations des routeurs
- de stocker ces informations dans la base de données
- de fournir aux clients la liste des routeurs disponibles

J'ai choisi d'intégrer MariaDB afin de travailler sur un véritable système de gestion de données, comme demandé dans le cahier des charges.

Une interface graphique me permet également de visualiser les routeurs connectés.

Routeur

Le routeur agit comme un relais intermédiaire.

Il :

- écoute sur un port spécifique
- se déclare automatiquement auprès du master
- reçoit un message et le transmet au routeur suivant ou au client final

J'ai développé une interface graphique pour simplifier la création des routeurs et éviter la multiplication des commandes en ligne.

Client

Le client permet :

- de se connecter au master
- de récupérer la liste des routeurs
- de sélectionner un chemin
- d'envoyer un message vers un destinataire

L'interface graphique du client centralise toutes ces fonctionnalités et facilite les tests sur plusieurs machines.

3. Protocole de communication

Pour simplifier le développement et le débogage, j'ai choisi un protocole textuel simple.

Déclaration d'un routeur

ROUTER <ID> <IP> <PORT> <E> <N>

Requête d'un client

CLIENT

Réponse du master

<ID> <IP> <PORT>

...

END

Routage d'un message

ROUTE|IP|PORT|MESSAGE

Chaque routeur traite uniquement la partie qui le concerne avant de transmettre le message.

4. Algorithme de chiffrement : principe, forces et faiblesses

Principe envisagé

J'ai cherché à mettre en place un système inspiré du routage en oignon (Tor).

L'idée était :

1. que chaque routeur génère une paire de clés RSA
2. que la clé publique soit transmise au master
3. que le client chiffré une clé symétrique avec RSA
4. que le message soit chiffré couche par couche

Forces

Même si le chiffrement n'est pas totalement fonctionnel, ce travail m'a permis :

- de comprendre le fonctionnement du RSA
- de manipuler des clés publiques et privées
- de mieux appréhender les problématiques de sécurité réseau
- de réfléchir à la confidentialité des communications

Faiblesses

La mise en œuvre s'est révélée plus complexe que prévu :

- gestion des tailles de clés délicate
- sérialisation des données difficile
- débogage très compliqué sur un système distribué
- nombreuses erreurs lors des tests sur plusieurs machines

Le temps consacré à cette partie a fortement impacté l'avancement global du projet.

5. Rapport de projet et gestion du projet

Organisation du travail

J'ai réalisé ce projet en autonomie, depuis l'analyse du cahier des charges jusqu'à l'implémentation finale.

J'ai dû :

- concevoir l'architecture
- écrire l'ensemble des scripts
- gérer les tests
- adapter le projet lors du passage du local à un environnement distribué

Gestion du temps

Le diagramme de GANTT initial n'a pas été respecté.

La principale raison est le temps important que j'ai consacré :

- à la compréhension du chiffrement
- à la correction de nombreux bugs
- au remaniement complet des scripts lors du passage sur plusieurs machines

Bilan personnel

Ce projet m'a permis d'apprendre beaucoup de choses, notamment :

- la programmation réseau en Python
- l'utilisation de MariaDB
- la création d'interfaces graphiques avec PyQt5
- le déploiement d'une application sur plusieurs machines
- Le principe de Thread

Même si certaines fonctionnalités ne sont pas finalisées, je considère ce projet comme très formateur.

Conclusion

En conclusion, cette SAE m'a permis de développer des compétences techniques importantes et de mieux comprendre les contraintes d'un projet informatique réel.

Malgré les difficultés rencontrées, notamment autour du chiffrement, le projet reste en partie fonctionnel (communication), documenté et cohérent.