

RE213 - Projets de Programmation Réseaux

R&I 2024-2025

Dans le cadre du module RE213 vous avez été exposés aux paradigmes de programmation réseaux avec l'API `socket`. Vous avez pris en main les appels systèmes nécessaires pour la réalisation d'un client-serveur TCP permettant de gérer en simultané plusieurs connexions établies. Vous avez également vu les protocoles HTTP et P2P. Vous allez implémenter un projet réseau visant à consolider les acquis, les mettre en pratique et acquérir des concepts avancés de programmation réseaux.

Organisation

Ces projets sont volontairement peu guidés et relativement libres, vous allez concevoir votre propre application réseau, établir des user stories, réaliser son architecture logicielle, documenter son API et développer votre projet.

Chaque projet doit être réalisé en binôme et doit être développés sur le volume horaire des séances restantes. Les livrables attendus sont les suivants:

- Code source complet + un guide pour installer et exécuter le système(README) ainsi que pour expliquer la manière dont les utilisateurs peuvent interagir avec le service.
- Un rapport exposant les fonctionnalités implémentées, les choix de conception, ainsi que le protocole de communication de la couche applicative (requêtes/réponses, sémantique et syntax des messages, règles de traitement, objets transférés etc.).
- Une démonstration de l'application.

Les projets seront évalués sur la base des critères suivants:

- Qualité du code (structure, lisibilité, modularité, documentation)
- Utilisation correcte des appels systèmes en lien avec les `sockets` réseau (read/write, accept, connect, etc.).
- Degrés d'achèvement du projet et fonctionnalités implémentées
- Qualité de la documentation
- Qualité de la démonstration/présentation orale (15 minutes).

Sujet 1 : Application de messagerie instantanée avec salon de discussion et partage de fichiers

Objectif et description : L'objectif de ce projet est de développer une application de messagerie instantanée avec un système de salon de discussion et un partage de fichiers via un serveur central. Ce type de projet combine plusieurs aspects essentiels de la programmation réseau et de la gestion des utilisateurs, avec une application pratique : un service de communication en temps réel, similaire à des plateformes comme Slack, Discord ou IRC. L'application doit fournir des fonctionnalités de base comme la possibilité de rejoindre un salon de discussion (par exemple, un canal thématique), d'envoyer des messages en temps réel, et de partager des fichiers avec les autres utilisateurs du salon. Les objectifs de ce projet sont les suivants : 1. Permettre à plusieurs utilisateurs de se connecter à un serveur, de participer à des salons de discussion, et d'échanger des fichiers en temps réel. 2. Gérer l'authentification des utilisateurs, les droits d'accès et la sécurité des échanges. 4. Offrir une expérience interactive en temps réel avec des fonctionnalités de partage et de gestion de fichiers.

Exigences techniques : Gestion et authentification des utilisateurs. Salons de discussion. Messagerie en Temps Réel. Partage de fichiers. Gestion des droits d'accès sur les fichiers partagés.

Extensions possibles :

- Sauvegarde des discussions.
- Synchronisation multi-serveurs.
- Gestion des collisions de versions de fichiers.
- Implémentation de transferts sécurisés avec SSL/TLS.
- Utilisation de protocoles de contrôle de flux pour améliorer les performances des transferts.

Sujet 2 : Serveur de jeu en ligne multijoueur

Objectif et description : L'objectif de ce projet est de développer un serveur de jeu en ligne multijoueur où plusieurs joueurs peuvent se connecter et interagir en temps réel. Ce type de projet combine plusieurs aspects importants de la programmation réseau, comme la gestion des connexions simultanées, la synchronisation des états des joueurs, et la communication entre clients et serveurs. Le jeu devra suivre un modèle client-serveur, dans lequel le serveur gère la logique du jeu et synchronise les actions des joueurs.

Il faudra développer un serveur capable de gérer un jeu multijoueur en ligne. Le jeu peut être un quiz, un tic-tac-toe, un pendu, un jeu de stratégie tour par tour, ou tout autre jeu simple que l'on peut concevoir dans un premier temps. Chaque joueur se connectera à un serveur central qui synchronisera les actions des différents joueurs et garantira que le jeu se déroule de manière cohérente et juste pour tous. Une liberté est laissée pour choisir les règles et la structure du jeu, ainsi que les mécanismes de communication. Le serveur sera responsable de la logique du jeu, tandis que les clients transmettront les actions des joueurs au serveur et recevront des mises à jour sur l'état du jeu.

Exigences techniques : Interface utilisateurs, Authentification des utilisateurs, Gestion des connexions client. Synchronisation des états des joueurs. Gestion des actions des joueurs dans un environnement temps réel. Logique de jeu Possibilité d'ajouter un chat en jeu et des classements.

Extensions possibles :

- Sauvegarde des parties pour permettre la reprise.
- Classements et statistiques de jeu
- Optimisation des échanges pour minimiser la latence.

Sujet 3 : Proxy réseau avec filtrage et journalisation

Objectif et description : Le projet consiste à développer un proxy réseau qui intercepte, filtre et journalise les connexions et messages envoyés sur le réseau, en particulier les connexions HTTP ou TCP. Les proxys sont couramment utilisés dans les environnements d'entreprise pour améliorer la sécurité réseau, surveiller le trafic, et appliquer des politiques d'accès aux ressources Internet. L'objectif de ce projet est de créer un proxy capable de : 1. Filtrer certaines connexions réseau en fonction de règles prédéfinies (comme bloquer l'accès à certains domaines, mots-clés, ou types de fichiers). 2. Journaliser toutes les connexions en enregistrant des informations telles que l'heure de la requête, l'adresse IP de l'utilisateur, et l'URL demandée. 3. Rediriger les connexions valides vers les serveurs cibles après vérification des règles de filtrage.

Le proxy doit se comporter comme un intermédiaire entre les clients (navigateur ou autres applications réseau) et les serveurs (serveurs web, serveurs applicatifs). Lorsqu'un client envoie une requête à un serveur, le proxy intercepte cette requête, la traite, puis la relaie ou la bloque en fonction des règles définies. Les fonctionnalités de base incluent : le filtrage (Bloquer les requêtes vers des domaines interdits, ou celles qui contiennent certains mots-clés (dans le cas d'HTTP)), la journalisation (enregistrer

des informations détaillées sur les requêtes réseau interceptées dans un fichier de log) et la redirection (si une requête est validée, elle doit être transmise au serveur cible pour réponse).

Exigences techniques : Interception et analyse des paquets (HTTP ou TCP). Filtrage des connexions basé sur une liste de règles (fichiers de configuration). Journalisation des connexions avec détails (heure, IP, URL).

Extensions possibles :

- Ajout d'un système de contrôle parental.
- Ajout d'un cache
- Gestion des connexions sécurisées (HTTPS).

Sujet 4 : Kamikaze

Aucun sujet ne vous plait ? Alors à vous de proposer un sujet à faire valider par l'enseignant avant de passer à sa réalisation !