

## Conception et architectures des réseaux

### Projet applicatif

## Messagerie instantanée interactive sur le réseau local

### Objectif

Il s'agit d'un projet réalisé en équipe et qui vise la conception d'un protocole et son implémentation dans un logiciel pour la communication interactive sur le réseau local. Le but principal est de matérialiser vos connaissances en programmation réseau sous forme d'un logiciel qui permettra d'organiser des échanges interactifs de données entre les utilisateurs qui se trouvent sur le même réseau. Le langage de programmation conseillé est Java ou C/C++.

### Description du projet

La messagerie instantanée (chat) est devenue un outil indispensable de communication interactive et est aujourd'hui largement appréciée par les utilisateurs du monde entier. Plusieurs fournisseurs de services sur le web proposent aujourd'hui leurs clients de messagerie instantanée basés soit sur leurs propres protocoles (propriétaires), parmi eux Yahoo Messenger, Windows Live Messenger, soit sur des protocoles ouverts (tels que IRC, SIP, XMPP, MQTT, etc.) : Pidgin, Empathy, Google Talk, Facebook Messenger, etc..

On vous propose de travailler sur un projet de conception et d'implémentation d'un système de messagerie instantanée. Ce projet inclut deux parties principales :

- la définition d'un cahier de charges et la conception de votre propre protocole de messagerie instantanée (la version la plus simple possible);
- l'implémentation de ce protocole dans un logiciel pour la communication interactive sur un réseau local.

Les conditions de fonctionnement et les fonctionnalités de base de votre messagerie:

- les postes des utilisateurs se trouvent sur un réseau local (donc sur le même réseau IP) ;
- les utilisateurs sont nomades, ils ne conservent pas leurs adresses IP et peuvent se connecter à partir de postes différents ;
- la gestion de votre système est décentralisée, donc pas de serveur central pour gérer les utilisateurs et les sessions d'échange. Elle doit être réalisée d'une façon totalement distribuée sur les postes clients en s'appuyant sur le protocole dont vous allez concevoir ;
- la communication sera faite par des messages asynchrones sous forme de chaînes de caractères de longueurs différentes ;
- on a un seul groupe de discussion sur le réseau ;
- chaque utilisateur doit s'enregistrer sur le réseau lors de sa connexion en fournissant un identifiant (pseudonyme);
- chaque poste doit maintenir une liste des utilisateurs connectés au système de messagerie (gestion dynamique des connexions/déconnexions) ;
- la gestion des états de l'utilisateur (connecté, absent, occupé, etc.) ;

L'interface utilisateur doit être minimale (terminal textuel dans le cas le plus simple)

Vous pouvez apporter des améliorations à cette version basique. Néanmoins, il est important que vous disposiez d'abord de programmes réalisant correctement ces fonctionnalités de base.

Voici quelques fonctionnalités supplémentaires possibles:

- plusieurs groupes de discussion ;
- chat à plusieurs (conférence) ;
- échange/partage de fichiers ;
- etc. .

Pour avoir une vue globale sur la définition de telles types de protocoles et pour vous inspirer, regardez la description simplifiée du protocole Yahoo Messenger [1].

L'idée de base de votre projet est la conception d'un protocole et le codage d'un logiciel pour faire des échanges interactifs sous forme de messages textuels, donc faire du clavardage (ou du chat, en anglais). A vous de définir le format de messages, leurs sémantique, les commandes, les procédures, etc. qui permettront d'organiser ce système et d'assurer son bon fonctionnement et sa robustesse. Ça sera de même à vous de faire le choix d'un protocole de transport qui correspond au mieux aux besoins de votre système de messagerie interactive.

Après avoir conçu le protocole et l'avoir implémenté sous forme d'un logiciel pensez à proposer un scénario le plus complet possible pour le tester afin de montrer toutes ses fonctionnalités et toutes ses parties intéressantes ainsi qu'à

évaluer ses performances.

## Évaluation du projet

L'évaluation de ce projet consiste de deux parties :

- un rapport détaillé sur l'ensemble de vos travaux sur ce projet (cahier de charge, phase de conception, phase d'implémentation, phase de test) ;
- une présentation de 20 minutes devant la classe accompagnée d'une démonstration du fonctionnement de votre logiciel.

Les présentations seront organisées pendant la dernière séance de cours (date à préciser dans l'emploi du temps). Mettez l'accent surtout sur la description détaillée de protocole dont vous avez conçu. Chaque présentation sera suivie de réponses aux questions et des discussions.

### Le contenu du rapport

- ✓ La présentation du protocole et des logiciels que vous avez conçus (le scénario, les algorithmes, les formats de données, les commandes, les types de messages, etc.) ;
- ✓ Le code documenté de votre programme, le mode d'emploi, les résultats obtenus ;
- ✓ Vos conclusions.

## Barème de correction<sup>1</sup>

Votre projet final sera évalué et noté selon les critères suivants:

<b>Exposé du problème et documentation</b> <i>Donnez une brève description de votre projet, ses limites et les instructions nécessaires pour le compiler et l'utiliser sous forme d'une petite documentation</i>	1 point
<b>Discussion</b> <i>Description détaillée de votre projet: fonctionnement, efficacité, difficultés (particularités) rencontrées, limites, algorithmes, organigrammes, etc. ... Exactitude des opérations, l'interface utilisateur, etc.</i>	3 points
<b>Tests et évaluation de performance</b> <i>Décrivez les tests que vous avez fait et donnez les détails de vos expériences et les résultats de mesure.</i>	1,5 points
<b>Perspectives</b> <i>Discussions sur des éventuelles extensions fonctionnelles de votre projet (protocole et logiciel) .</i>	0,5 points
<b>Conclusions et recommandations</b> <i>Évaluation des résultats, explications, améliorations, recommandations, atteinte des objectifs, etc. ...</i>	1 point
<b>Qualité du logiciel</b> <i>Code commenté, bien structuré, lisible, etc. ...</i>	2 points
<b>Présentation et qualité du français</b>	1 point

**Des crédits supplémentaires** allant jusqu'à 2 points pourront être attribués si vous mettez en œuvre des fonctionnalités supplémentaires intéressantes et utiles.

Vous devez remettre vos résultats sous forme d'une archive contenant votre rapport (format pdf), votre code source avec des fichiers complémentaires nécessaires pour compiler (le makefile) et lancer votre application et avec des instructions bien précises sur l'utilisation de votre logiciel.

La date limite de remise des rapports et des documents dans le dossier est précisée sur la page de module.

Tout retard sera pénalisé de 1 point par jour de retard.

## Références

1. Yahoo Messenger Protocol v9 <http://libyahoo2.sourceforge.net/ymsg-9.txt>
2. MQTT Protocol, version 3.1.1, <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>

---

<sup>1</sup> Les poids sont à titre indicatif