



**École Polytechnique de Montréal**  
 Département Génie Informatique et Génie Logiciel  
 INF3405 – Réseaux Informatiques

**TP4 : Projet**

**Système de questionnaire interactif**

**1. Informations générales**

Session	Été 2017
Public cible	Étudiants de 1 <sup>er</sup> cycle
Date et Lieu de réalisation	À partir du 30 mai 2017 au Laboratoire L-4708
Taille de l'équipe	2 étudiants
Pondération	10 %
Date de remise du projet	13 juin 2017 (23h55 au plus tard)
Directives particulières	<ul style="list-style-type: none"> <li>✓ Tout rapport sera pénalisé de <b>5</b> points s'il est soumis par une équipe dont la taille est différente de deux (02) étudiants sans l'approbation préalable du chargé de laboratoire.</li> <li>✓ Soumission du rapport (en format PDF) par <b>moodle</b> uniquement (<a href="http://moodle.polymtl.ca">http://moodle.polymtl.ca</a>).</li> <li>✓ <b><u>Soumission d'une archive compressée contenant le rapport et le code source de l'application, et éventuellement les exécutables.</u></b></li> <li>✓ Toute soumission de l'archive en retard est pénalisée à raison de 3 points par jour de retard.</li> </ul>
Chargé de laboratoire	Fabien BERQUEZ ( <a href="mailto:fabien.berquez@polymtl.ca">fabien.berquez@polymtl.ca</a> )
Auteur :	Fabien BERQUEZ

**2. Connaissances requises**

- Sockets
- Threads
- Programmation C++ ou Java

### 3. Objectifs du laboratoire

L'objectif de ce laboratoire est de familiariser l'étudiant :

- aux échanges Client/Serveur en utilisant les sockets en mode connecté ;
- au développement « d'applications réseau » en utilisant les threads.

### 4. Description

Dans le cadre d'un cours donné par un professeur, celui-ci décide de vous confier la conception et le développement d'une application (un client et un serveur) pour lui permettre d'interagir avec les étudiants. L'idée qu'il se fait de l'application est la suivante : il veut pouvoir, lors du déroulement du cours, poser des questions aux étudiants et collecter leurs réponses, afin de s'assurer que la compréhension est bonne et pouvoir reprendre les explications sur des points qui ne seraient pas compris. Le serveur, qui serait exécuté sur le poste du professeur, lui permettrait de poser la question et de voir les réponses qui ont été proposées. Le client, exécuté par chacun des étudiants sur leur propre machine, permettrait de se connecter au serveur, de réceptionner la question, et d'envoyer une proposition de réponse.

Au démarrage du serveur, celui-ci demande à l'utilisateur (le professeur) d'entrer les informations suivantes : adresse IP du poste, port d'écoute (un port entre 6000 et 6050) et durée du « sondage » (en secondes). Le serveur propose alors à l'utilisateur de définir une question (d'une taille qui ne dépasse pas les 200 caractères). Le décompte commence alors.

Au lancement du client, celui-ci demande à l'utilisateur d'entrer l'adresse IP du serveur, et le port du serveur. Ces informations sont utilisées pour tenter de se connecter au serveur. Le serveur vérifie alors qu'il reste du temps avant l'expiration du délai fixé, et accepte la participation dans ce cas. Sinon, il n'autorise pas le client à participer, et le client affiche « Le sondage a expiré ». Si la participation a été acceptée, le serveur envoie la question au client. Le client demande à l'utilisateur de donner sa réponse (taille ne dépassant pas les 300 caractères), puis l'envoie au serveur. À la réception de cette réponse, le serveur affiche l'adresse IP et le port sources de la réponse ainsi que la réponse en elle-même.

Notez que le programme client et le programme serveur ne sont (en théorie) pas exécutés sur la même machine et que, bien évidemment, plusieurs clients peuvent tenter de se connecter au serveur en même-temps. Cependant, chaque client connaît d'avance l'adresse IP du serveur ainsi que le numéro de port écouté. Durant le « sondage », le serveur affiche et stocke dans un fichier au fur et à mesure les informations mentionnées précédemment : adresse IP, port utilisé et réponse. Par exemple

132.207.29.107 : 56887 - Réponse de l'étudiant 1

132.207.29.115 : 54563 - Réponse de l'étudiant 2

## 5. Requis fonctionnels

Les fonctionnalités attendues sont indiquées comme suit :

### ✓ Serveur

- Saisie des paramètres du serveur (adresse IP, port d'écoute entre 6000 et 6050, durée du sondage)
- Saisie de la question du sondage (200 caractères maximum)
- Ouverture du sondage : À cette étape, le serveur se met en mode écoute sur le port sélectionné et lance le décompte du sondage
- A chaque requête de connexion au port écouté, vérifier que le sondage est toujours actif. Si oui, transmettre la question, sinon notifier le client que le sondage est terminé.
- Recevoir les réponses des clients
- Tenir un journal des réponses. Il s'agit d'un fichier nommé *journal* au format .txt édité par le serveur, qui comprend sur chaque ligne l'adresse IP, le port source et la réponse associée.
- Fermeture du sondage : À l'expiration du temps alloué au sondage, les nouvelles connexions se voient notifier que le sondage est terminé. Les connexions encore en cours peuvent être soit poursuivie, soit coupées, selon votre préférence.)
- Afficher au cours de l'exécution les réponses reçues (adresse IP, port source, réponse)

### ✓ Client

- Saisir au clavier l'adresse IP du serveur et le port sur lequel le serveur écoute.
- Vérifier la validité de l'adresse IP saisie (uniquement le format) et le numéro de port (entre 6000 et 6050)
- Se connecter au serveur.
- Réceptionner la question ou Afficher un message « Le sondage a expiré » selon la situation.
- Saisir une réponse (300 caractères maximum).
- Transmettre la réponse au serveur.
- Se déconnecter.

## 6. Livrable

### 1. Langages et bibliothèques autorisés

Le client et le serveur doivent être développés dans un des langages suivants :

- C/C++
- Java

Dans le cas d'un développement en C/C++, les bibliothèques utilisées peuvent être :

- WinSock 2
- Qt (par exemple module Qt Network).

**De manière générale, les projets doivent pouvoir être exécutés sur les postes du laboratoire L4708 sans configuration supplémentaire (i.e tels que remis sur Moodle).**

Tout non-respect de ces consignes pourra entraîner une pénalité pouvant atteindre **10 points sur 20**.

### 2. Soumission

Le livrable est une archive (ZIP ou RAR) dont le nom est formé des numéros matricules des membres de l'équipe, séparé par un trait de soulignement ( \_ ). Votre archive contiendra les fichiers suivants :

- Les fichiers sources (.java, .cpp, .h, ...)
- les fichiers projets (solution Visual Studio, projet Qt (.pro), projet Eclipse, ...)
- le rapport au format PDF ou Word

*Non obligatoire mais vivement conseillé* : les fichiers exécutables (.exe)

### 3. Rapport

Le rapport doit comporter les éléments suivants :

- **Page présentation** qui doit contenir le nom ou le logo de l'école, le libellé et l'identifiant du cours, la session, le numéro et l'identification du projet, la date de remise, les matricules et noms des membres de l'équipe, la mention « Soumis à : **nom et prénoms du chargé de laboratoire** ».
- **Introduction** avec vos propres mots pour mettre en évidence le contexte et les objectifs du TP.
- **Présentation** de vos travaux. Une explication de votre solution mettant en lumière la prise en compte des principaux requis du système. Si vous utilisez des configurations particulières des bibliothèques ou des projets, précisez-les également.
- **Difficultés rencontrées** lors de l'élaboration du TP et les éventuelles solutions apportées.
- **Critiques et Améliorations** : Il serait intéressant d'inclure vos suggestions pour améliorer le laboratoire.

- **Conclusion :** Expliquez-en quoi ce laboratoire vous a été utile, ce que vous avez appris, si vos attentes ont été comblées, etc.

## 7. Évaluation

Évaluation de l'exécutable	<b>6</b>
Évaluation de l'implémentation: gestion adéquate des variables et de toute ressource (création, utilisation, libération), gestion des erreurs, logique de développement, documentation du code, etc.	<b>8</b>
Rapport	<b>6</b>
<b>Total des points</b>	<b>20</b>