

Programmation réseaux

TP 8

Programmation TCP en Java — Peer to peer

Mars 2007

Dans ce TP, nous allons programmer en Java un jeu d'allumettes en peer to peer.

1 Règles du jeu

Le jeu se joue à n joueurs, qui jouent à tour de rôle. Au départ, m allumettes sont disposées sur la table. Lorsque vient son tour, un joueur doit retirer entre 1 et k allumettes. Le joueur qui prend la dernière allumette a perdu.

2 Le protocole

Nous allons implémenter ce jeu en utilisant un protocole pair-à-pair avec état répliqué. Ici l'état est un entier (le nombre d'allumettes restantes). Nous supposons que chaque pair connaît tous les autres et l'on fixera un ordre de jeu. Au démarrage du jeu, nous établirons des connexions entre tous les pairs, qui resteront ouvertes pendant toute la durée du jeu. Ensuite les pairs jouent à tour de rôle en envoyant à tous les autres le nombre d'allumettes qu'ils retirent.

Le protocole utilisé en phase de jeu est donc très simple : les seuls messages envoyés sont des chaînes de caractères représentant des entiers.

3 Implémentation

Implémenter une première version du programme en utilisant le protocole décrit ci-dessus. On supposera que les pairs respectent scrupuleusement les règles et qu'ils utilisent exactement les mêmes valeurs de k et m .

Pour simplifier la première version, le programme connaîtra par avance les adresses IP et numéros de ports de tous les joueurs et l'ordre de jeu. Il prendra en paramètres le rang du joueur dans l'ordre de jeu, puis m et k .

Voici une stratégie possible pour établir toutes les connexions.

- Chaque joueur commence par attendre la connexion du dernier (`accept()`) puis un message de celui-ci indiquant qu'il a fini de se connecter à tout le monde.
- Ensuite il fait la même chose avec l'avant-dernier, et ainsi de suite pour tous les joueurs qui jouent après lui.
- Ensuite il essaie de se connecter à ceux qui sont avant lui (`new Socket(addr, port)`), dans l'ordre décroissant, en répétant la tentative toutes les secondes en cas d'échec.
- Ensuite il prévient tout le monde qu'il a fini ses connexions.
- Pour terminer il attend que les plus petits aient fini de se connecter (en faisant des `readLine()` dans l'ordre décroissant).