

Programmation d'application distribuée et en réseau

Projet

Fabrice BODSON

Florian NICOLAS

Lucas THITEUX

Thibaut WATRISSE

Thomas COLLIGNON

Justin ROLLAND

1.	Introduction	2
2.	Principe de base de l'application	2
2.1.	Réseau applicatif	2
2.1.1	Topologie.....	2
2.2	Création d'un réseau applicatif	3
2.3	Ajout d'un client dans le réseau applicatif.....	3
2.3.1	Déconnexion d'un client dans le réseau applicatif	3
2.4	Partage de fichiers.....	3
2.4.1	Listing des fichiers.....	3
2.4.2	Organisation des partages	3
2.4.3	Transfert d'un fichier	3
2.5	Expérience utilisateur	4
2.5.1	Interface	4
2.5.2	Listing des fichiers.....	4
2.6	Fusion des réseaux (merge)	4
2.7	Chiffrement des transferts	4
2.8	Priorité des transferts	4

1. Introduction

Dans le cadre de notre projet de programmation d'application distribuées et en réseau, nous avons dû développer une application permettant de partager des documents entre les différentes instances de notre application de manière décentralisée.

Afin de bien comprendre les principes et fonctionnements de l'application, nous allons partir du principe que nous disposons d'un réseau applicatif composé de trois clients, plus un quatrième souhaitant le rejoindre. Nous entendons par client une instance de l'application.

2. Principe de base de l'application

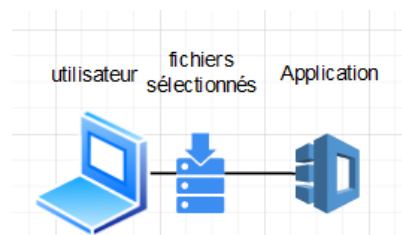
2.1. Réseau applicatif

Un réseau applicatif, est un ensemble de clients communiquant entre eux afin de partager des fichiers. On peut parler d'un réseau applicatif dès que deux clients interagissent entre eux.

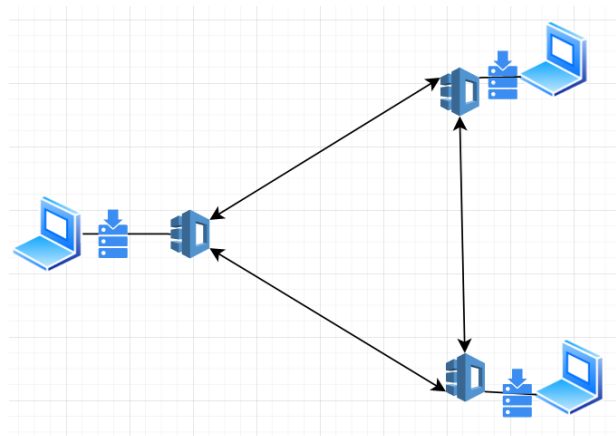
2.1.1 Topologie

Ici, un utilisateur sélectionne les fichiers qu'il désire mettre à disposition au travers de l'application.

Exemple « français » : je(utilisateur) décide de partager ces fichiers (fichiers sélectionnés), et je le dis à l'application.



Ensuite les clients s'interconnectent par l'intermédiaire de l'application :



Ainsi, les fichiers sont disponibles pour chaque client dans le réseau applicatif.

2.2 Création d'un réseau applicatif

Un client a la possibilité de créer un réseau applicatif lorsque ce dernier n'existe pas.

2.3 Ajout d'un client dans le réseau applicatif

Pour qu'un client rejoigne le réseau applicatif, il faut qu'il connaisse au moins l'adresse IP d'un client dans le réseau applicatif. Ainsi, il recevra les informations dont il a besoin et l'application communiquera son adresse IP aux autres clients afin qu'ils connaissent la présence du nouveau client dans le réseau applicatif. Le client devra également donner le port sur lequel il souhaite communiquer pour ainsi éviter les conflits de ports côté client.

2.3.1 Déconnexion d'un client dans le réseau applicatif

Lorsqu'un client veut quitter le réseau applicatif, l'application l'annonce à tous les clients afin que ceux-ci puissent supprimer son adresse de leur liste de pairs. Ensuite la machine quitte le réseau.

2.4 Partage de fichiers

Chaque machine met à disposition de l'application un ensemble de fichiers qui sont partagés. C'est-à-dire que, via l'application, ce sont des fichiers qui peuvent être téléchargés ou listés par un utilisateur.

2.4.1 Listing des fichiers

Un utilisateur peut lister les fichiers disponibles sur le système de fichiers distribués. Cette liste est obtenue par le client qui envoie dynamiquement des requêtes auprès des clients du réseau applicatif. Si plusieurs clients possèdent le même fichier, ce fichier ne sera affiché qu'une seule fois.

2.4.2 Organisation des partages

Afin d'éviter tout conflit dans les fonctionnalités décrites ci-après entre plusieurs clients disposants du/des même(s) fichier(s), ils se mettent d'accord pour qu'un seul d'entre eux mette sa copie du fichier à disposition du client ayant fait la requête en fonction de l'algorithme de priorité de transfert.

2.4.3 Transfert d'un fichier

Chaque client peut demander à télécharger un ou plusieurs fichier(s) auprès des clients qui disposent de ce fichier. Le client **demandeur** sera en lien direct avec le client choisi.

2.5 Expérience utilisateur

Un utilisateur utilise sa machine (= ordinateur) pour exécuter l'application qu'on développe (= devient un client car on crée une instance de l'application).

Un utilisateur peut, en passant par l'application, lister l'ensemble des commandes disponibles, lister les fichiers disponibles, télécharger le/les fichier(s) qu'il souhaite, il peut également rejoindre le réseau applicatif (ajouter sa machine) et se déconnecter.

2.5.1 Interface

L'utilisateur peut interagir avec l'application par le client. Celui-ci dispose d'un terminal à choix multiples sur lequel il peut exécuter différentes commandes.

2.5.2 Listing des fichiers

Au moyen d'une entrée dans le terminal, l'utilisateur peut consulter les fichiers disponibles à travers les clients du système de fichiers distribués.

2.6 Fusion des réseaux (merge)

Lorsque l'utilisateur va initier la fusion entre deux réseaux, il va devoir entrer les différentes informations nécessaires et ce afin de contacter un client de l'autre réseau. Celui-ci va alors envoyer sa liste de pairs au client qui l'aura contacté. Le client qui aura initié la fusion contactera ensuite chaque pair de cette liste reçue afin qu'ils s'ajoutent mutuellement dans leurs listes. Enfin, il communiquera cette nouvelle liste aux pairs présents dans son « ancienne » liste de pairs afin qu'eux aussi fusionnent. Le client contacté pour la fusion reçoit des demandes d'ajout à la liste de pairs comme si un nouveau client s'ajoutait au réseau.

2.7 Chiffrement des transferts

Le transfert des fichiers est chiffré au moyen du protocole SSL. Cela nécessite donc l'utilisation de certificats. Ce procédé se fera au moyen de certificat auto-signé. Dès lors, quand un transfert de fichiers se fait, les clients s'échangent leur certificat au préalable. La vérification est cependant bypassée.

2.8 Priorité des transferts

Lorsque plusieurs clients disposent d'un fichier demandé, le programme va regarder pour prendre au possible un client qui n'est pas déjà en train de faire un transfert actuellement. Si tous les clients détenant le fichier sont occupés à transférer, le choix est aléatoire.