Projet de Réseau
Version centralisée

Song Chen
Rouxel Quentin
Abid Rachid
Gueye Papa Magueye

23 mars 2012

Introduction

Chapitre 1

Tracker

Chapitre 2

Pair

2.1 Motivations du choix du langage : Java

Clairement, le plus gros du travail d'implémentation de ce projet se situe au niveau du client. Le client a d'une part les fichiers sur le disque à gérer mais également plus de messages différents dans le protocole à implémenter. Pour cette raison, nous avons préféré utiliser pour le développement du pair un langage de haut niveau tel que Java afin de nous simplifier la tâche et de laisser l'emploi du C au tracker, plus simple.

2.2 Utilisation du buffermap

Pour que le pair puisse connaître les pièces disponibles d'un fichier temporaire, un buffermap est mis en place dans la structure de chaque fichier. Il est une séquence de bits dont le nombre correspond au nombre des pièces du fichier. Chaque bit indique la présence de la pièce correspondante dans le buffer.

En considérant le nombre de bits par rapport à la taille d'un fichier, le buffermap dans notre structure est implémenté comme un tableau. On constate ensuite que pour une pièce de fichier, il existe trois états différents pour lesquels on doit effectuer différents traitements :

- soit cette pièce n'existe pas localement. Dans ce cas là, le pair demande à son voisin leur disponibilité sur cette pièce.
- soit cette pièce est en train de téléchargée. Donc le pair n'a pas besoin de faire le demande sur cette pièce mais il ne peut pas non plus de fournir des données de cette pièce, car ces dernières ne sont pas complètes.
- soit elle est finie de télécharger ou est déjà disponible dans le disque. Elle est prêt d'être envoyée au voisin.

Pour la raison qu'on doit coder ces 3 cas dans une case de tableau, un bit binaire n'est pas suffisant. C'est pourquoi on définit un tableau d'entier.

2.3 Détection de la fin de transmission

Une autre problématique technique est de détecter la fin d'une transmission de message. En effet, le protocole autorise à transférer des données binaires. Il est alors impossible de convenir d'un symbole correspondant à la fin de la transmission. Puisque le protocole n'indique pas la taille des données binaires, il faut à partir des données connues calculer la taille des données à lire. De ce fait, la détection de la fin de la transmission est spécifique à chaque message du protocole.

2.4 Stockage des fichiers temporaires

Le gestions des données temporaires, c'est à dire des pièces d'un fichier en cours de téléchargement est également un problème difficile. Dans une première approche, nous pensions écrire dans un répertoire approprié sur le disque chaque pièces dans un fichier séparé. Cependant, vu la taille réduite de chaque pièce et le grand nombre de pièces potentielles pour les grand fichiers (plusieurs millions) nous avons choisi une autre approche afin de ne pas saturer le système de fichier.

Un unique fichier temporaire est créé par fichier en cours de téléchargement. Les pièces sont écrites les unes à la suite des autres en binaires dans ce fichier (pas dans l'ordre des données mais dans l'ordre de réception des pièces).

Bien entendu, ces fichiers comporte un header spécifique regroupant les informations essentielles du fichier en cours de téléchargement (taille, clef et taille des pièces) mais également la disposition des pièce dans le fichier.

Conclusion