Down with hierarchy: File-sharing with CP2PC

Ihor Kuz Patrick Verkaik Maarten van Steen Arno Bakker

Vrije Universiteit
Amsterdam, The Netherlands

1 Introduction

There are currently numerous peer-to-peer (P2P) networks used for sharing files between clients. Although the various networks generally offer users similar functionality (e.g., searching for files, downloading files, etc.), each such network has its own suite of protocols for communicating between peers and sharing and distributing its files. As such, each network requires the use of its own proprietary client-side software. A user wishing to make use of several of these networks must therefore run several different client applications. Given the similar functionality offered by most such applications, this leads to much repetition of work and frustration.

In the cp2pc (pronounced copy to pc) project we tackle this problem in two steps. First we define a unified programming interface to P2P file-sharing systems. Second we use this programming interface to build a single unified P2P file-sharing client.

2 P2P File-Sharing Networks

Although the various file-sharing networks offer similar functionality at the user interface level, the lower level details of these networks differ widely. To get a good overview of the various issues involved in P2P file-sharing a number of existing netowrks were studied. These networks were chosen such that they offered a wide variety of features and different network models. We looked at Gnutella, a popular decentralized file-sharing network with broadcast based routing and many different implementations available; mnet, a file-sharing network evolved from Mojo Nation with a credit based file publishing network; jnushare, an application based on Sun's JXTA P2P framework; CFS, which provides a distributed read-only file system based on the Chord P2P research system and GDN, a Globe based software distribution network.

3 A File-Sharing Model

Given the differences between the available file-sharing networks, a unified programming interface must provide a single file-sharing model that can be mapped onto each of the individual networks. The basic functionality shared by all the networks studied includes:

- Files can be published to the network. A published file is available to be downloaded by other peers.
- Files can be downloaded from the network
- Clients can easily join and leave the network.
- Many of the systems also offer the capability of searching for a file given keywords or attributes.

Besides this functionality, all systems can be configured and monitored to some extent, however, the configuration parameters and monitoring details differ widely between systems.

4 A File-Sharing API

The file-sharing API that we've developed is based on the concept of files. Files can be published, downloaded and searched for. The API also offers functions that allow networks to be configured as well as monitoring of network status and file uploads and downloads. The API is defined in C, however it has been designed so that it can be implemented in a variety of other languages, including Java and C++.

5 A File-Sharing Application

The user application presents a typical file-sharing application interface. It allows files to be selected for sharing (aka publishing), it allows files to be searched for and it allows files to be downloaded. The application can use multiple P2P file-sharing networks simultaneously as shown in Figure 1.

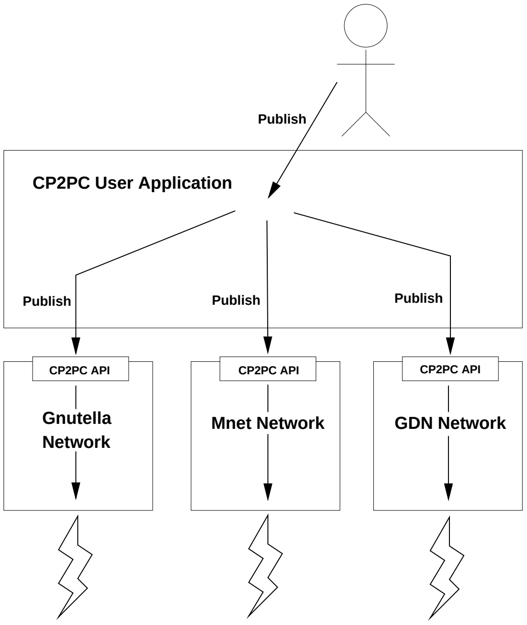

Figure 1: The CP2PC unified file-sharing client application

In this figure, a user request (in this case a publish request) is passed on to each supported network through the unified API. Each network performs the function in its own way. The results are returned to the application, grouped and shown to the user.

A The Authors

Of the following, Patrick and Ihor are the active developers, Maarten and Arno are simply lazy spectators.

Ihor Kuz is currently finishing his PhD at the TU Delft and Vrije Universiteit Amsterdam. He is researching Wide Area Distributed Systems and the World Wide Web as part of the Globe project. He is also working as a scientific programmer at the VU where he is working on P2P systems.

Patrick Verkaik is working as a scientific programmer for the Globe project at the Vrije Universiteit Amsterdam. He is the chief programmer for the Globe distributed system, and is also working on P2P systems.

Maarten van Steen is professor of computer science at the Vrije Universiteit Amsterdam where he has been leading the Globe project together with Andrew Tanenbaum. His research interests are mainly in large-scale distributed systems, including computer networks and Web-based systems.

Arno Bakker is currently a post-doc in the Globe project at the Vrije Universiteit in Amsterdam, and finishing his PhD on a worldwide distribution network for free software built using Globe technology.