cp2pc: A Component-based Peer-to-Peer Client*

Maarten van Steen

April 19, 2002

Problem There are currently several peer-to-peer (P2P) networks that are used for sharing files between clients. Each network uses its own proprietary suite of protocols for file sharing and distribution, which leads to many different client applications that are difficult or impossible to integrate. Currently, each P2P network requires its own proprietary client-side software. There are no means to integrate the clients, for example, to allow a user to operate on several P2P networks simultaneously. In fact, there are now systems available that use the same technology but cannot be integrated unless special measures are taken. For example, the P2P systems Kazaa, Grokster, and Morpheus all make use of Fasttrack's P2P protocol suite [http://www.fasttrack.nu/].

Background When taking a step back, it is seen that current P2P systems share a lot of functionality:

- Clients can upload files to a (decentralized) storage system
- Files can be searched for and downloaded from that same storage system using keywords
- Clients can easily join and leave the network

Differences between the various systems mainly stem from the way security is incorporated. Systems such as Gnutella [http://gnutella.wego.com/], Napster [http://www.napster.com/], and Fasttrack-based products (Grokster [http://www.grokster.com/], Kazaa [http://www.kazaa.com/], Morpheus [http://www.musiccity.com]) pay very little or no attention to security. Other systems like Freenet [2] and Publius [9] concentrate on anonymous uploads, while Free Haven [3] in addition also aims at anonymous downloads.

The Globe Distribution Network (GDN) [1] can also be considered a peer-to-peer system, although its object servers are more heavy-weight and are less flexible when it comes to adding or removing them from the network. Clients, however, can come and go as they like. GDN offers security in the

*Pronounce as "copy to PC."

form of traceability of uploading clients, possibly using pseudonyms instead of directly traceable identities.

A next generation of P2P systems is currently being constructed in which attention concentrates on efficiently routing requests between peers to provide highly scalable solutions. Examples of such systems include Chord [8], Pastry [6], and Tapestry [10], which all originate from the academic research community. In contrast, Sun Microsystems are launching their JXTA specifications and reference implementation [http://www.jxta.org]. These systems are subsequently used to build large-scale decentralized storage systems, in which the storage system is built as an application on top of the core P2P network (see, e.g., PAST [7] and OceanStore [4, 5]). However, these next-generation P2P systems essentially implement the same functionality as the other file-sharing P2P systems, although they significantly improve efficiency and scalability.

Goal This project has two goals.

The first goal is to develop a minimal programming interface to file-sharing peer-to-peer systems that can support many different P2P protocols. This interface can then be used by different groups to develop client-side applications.

The second goal is to develop a relatively simple graphics-based client that can be used for different P2P protocols, and that can integrate file sharing across different networks.

Approach The project is divided into two parts. The current proposal covers the start phase and will result in a detailed proposal for the second phase.

During the first phase, we concentrate only on the selection of P2P protocols that we want our client application to support. This research involves installing and running various systems, and perhaps building small applications or making enhancements to existing ones. The goal is to obtain a deeper insight in the way each system actually works, and notably the working and interfacing to its protocol suite. While doing this research, an initial design of a common interface should surface. This design should allow us to obtain insight in the amount of work needed to implement various protocols.

Candidate protocols to consider for implementation include the following.

GDN: GDN already provides a file-sharing interface that we are familiar with. Also, GDN differs from other P2P systems in the sense that it uses *deterministic* protocols instead of protocols that make use of *probabilistic* routing. For example, peers in GDN are known in advance, in other systems they are often not.

Mojo Nation: The interesting aspect of Mojo Nation is that downloading is based earned credits. The more the user contributes to the network, the

more credit he gets. Contributions consist of offering disk space and providing content.

JXTA: JXTA is Sun's framework for P2P. It consists of (deliberately under-specified) protocol specifications for which a reference implementation that covers a large part of these specifications is available. Considering the expected popularity of JXTA, it is worthwhile considering it as a candidate.

Gnutella: Unlike Napster, Gnutella is a fully decentralized system for which a protocol specification is readily available (version 0.4). Gnutella has its own way for routing and searching nodes, mainly using expensive multicast-based techniques. There are many Gnutella software clients that we can inspect for ideas concerning our own client-side application development.

Pastry, Tapestry, Chord: These three protocols are very similar in the sense that they use probabilistic routing and searching techniques that allow a P2P network to scale to hundreds of thousands of nodes. All of them have distributed storage systems built on top that offer immutable file sharing systems.

Fasttrack: This system forms the basis for three new popular P2P systems: Morpheus, Kazaa, and Grokster. It appears to use a tree-based distribution protocol as well as standard interface for routing and searching files. It is yet unclear to what extent we can have access to protocol specifications or source codes.

The first phase should be completed in approximately 3 months, starting on January 1, 2002.

Other initiatives The Computer Systems group at the VU is currently involved in a proposal to set up a large-scale peer-to-peer system to obtain a better insight between the effect of routing between logical nodes and actual network proximity. In this project, we aim to use an existing core P2P routing system and build a measurement tool on top of that. Data gathered from measuring the system will be stored using a storage system implemented on top of the core P2P system as also discussed above. Participating sites include MIT (Frans Kaashoek), Rice University (Peter Druschel), and EPFL (Rachid Guerraoui).

We will strongly advocate that the programming interface resulting from the first phase is used for developing the measurement application. As an effect, a relatively large community of researchers will be able to use and test the proposed interface. It also means that (a subset of) this community will have to be involved in providing feedback on the interface design. This initiative will presumably be announced during the IPTPS workshop to be held in Cambridge, MA on March 7–8, 2002. Note that this initiative is independent

from this proposal. However, if the research community actually succeeds in building a large peer-to-peer network, then clearly the dissemination of the work we propose here to a large community of users will be much easier.

References

[1] A. Bakker, E. Amade, G. Ballintijn, I. Kuz, P. Verkaik, I. van der Wijk, M. van Steen, and A. Tanenbaum. "The Globe Distribution Network." In *Proc. Ann. Techn. Conf. (FREENIX Track)*, pp. 141–152, San Diego, CA, June 2000. USENIX.

[2] I. Clarke, O. Sandberg, B. Wiley, and T. Hong. "Freenet: A Distributed Anonymous Information Storage Service." In H. Federrath, (ed.), *Designing Privacy Enhancing Technologies*, volume 2009 of *Lect. Notes Comp. Sc.*, pp. 46–66. Springer-Verlag, Berlin, 2001.

[3] R. Dingledine, M. J. Freedman, and D. Molnar. "Free Haven." In A. Oram, (ed.), *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, pp. 159–187. O'Reilly & Associates, Sebastopol, CA, 2001.

[4] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, R. G. Dennis Geels, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao. "OceanStore: An Extremely Wide-Area Storage System." In *Proc. Nineth Int'l Conf. Architectural Support for Programming Languages and Operating Systems*, pp. 190–201, Cambridge, MA, Nov. 2000. ACM.

[5] S. Rhea, C. Wells, P. Eaton, D. Geels, B. Zhao, H. Weatherspoon, and J. Kubiatowicz. "Maintenance-Free Global Data Storage." *IEEE Internet Comput.*, 5(5):40–49, Sept. 2001.

[6] A. Rowstron and P. Druschel. "Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems." In R. Guerraoui, (ed.), *Proc. Middleware 2001*, volume 2218 of *Lect. Notes Comp. Sc.*, pp. 329–350, Berlin, Nov. 2001. Springer-Verlag.

[7] A. Rowstron and P. Druschel. "Storage Management and Caching in PAST, a Large-Scale, Persistent Peer-to-Peer Storage Utility." In *Proc. 18th Symp. Operating System Principles*, Banff, Canada, Oct. 2001. ACM.

[8] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications." In *Proc. SIGCOMM*, San Diego, CA, Aug. 2001. ACM.

[9] M. Waldman, A. D. Rubin, and L. F. Cranor. "Publius: A Robust, Tamper-evident, Censorship-resistant, Web Publishing System." In *Proc. Nineth Security Symposium*, pp. 59–72, Denver, CO, Aug. 2000. USENIX.

[10] B. Y. Zhao, J. Kubiatowicz, and A. D. Joseph. "Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing." Technical Report CSD-01-1141, Computer Science Division, University of California, Berkeley, Apr. 2001.