Towards Next-Generation Peer-to-Peer Systems

Magnus Kolweyh, University of Bremen, mag@tzi.de

December 4, 2004

Abstract

The Internet has changed from a statically, television-like web page space to a highly heterogeneous and extremely dynamic information platform. No other technology forced that development like Peer-to-Peer. This short paper will briefly summarize emerging trends and promising concepts for Peer-to-Peer systems and describe upcoming challenges for developers.

1 Introduction

A significant new paradigm for the Internet are Peer-to-Peer (P2P) systems. Those networks create most of the Internet network traffic today and regroup millions of users in new clusters on the network level. Those P2P systems often contain millions of nodes [RFI02], terabytes of data [KL04] and cause most of the total Internet traffic nowadays [MKL$^+$02]. Despite their enormous performance power P2P networks are often reduced to the file sharing context [MKL$^+$02]. This limitation leads to a P2P discussion which focuses on terms like copyright and ownership and overlooks the power such networks accommodate from a technical point of view.

2 Spotlight on Peer-to-Peer

Today's P2P applications can be clutched into three areas: File sharing, Distributed Computing and Communication. When we are looking for new P2P application fields we definitely miss advanced services that go beyond communication and sharing of data or computer power. In contrast to Napster [Nap], the first prominent example of P2P, today's systems do not anymore employ a single centrally stored index. Protocols of famous P2P Applications like Kazaa

[LRW03], [Neo] or Gnutella 2 [Gnu] use hybrid architectures with super peers or hubs to scale with the vast amount of users. Apparently P2P offers a lot of advantages compared to traditional client-server systems such as performance and cost reduction, though P2P developers are confronted with quite a lot new challenges that traditional Web-Developers never had to deal with. Scientists have build several interesting concepts for P2P networks, e.g. Anthill, a complex adaptive system, where peers are modeled as ants that are able to build huge powerful nests when they collaborate appropriately [Mon01]. However, the field of P2P is driven by applications and their success. It's therefor necessary to look at promising concepts for various P2P challenges as well as consider trends in popular file sharing networks. The next sections will briefly present such challenges and discuss emerging trends of today's P2P systems.

3 Challenges

3.1 Scalability

While a single name indexing server like central Napster is easy to attack for several kind of institutions, e.g. music and video industry, a few years ago Gnutella seemed to be the perfect solution with a complete decentralized network consisting of servant peers. Gnutella was based on a protocol, which can be briefly summarized as broadcasting queries with a fixed limited scope.While in the year 2000 everyone expected Gnutella to become even bigger then Napster, research [RFI02] and the dramatically deprivation of users both showed the weakness of that plain Gnutella protocol. The effect was the implementation of mostly hybrid protocols for popular protocols like Fasttrack [LRW03] and the adaption of already existing concepts from distributed and concurrent computing [SMK$^+$01, DBK$^+$01]. Today there are several scale-proven systems [Mon01, CSWH01, KBC$^+$00]. However, scalability in P2P also has to be proved in real-life situations. Beside the integration of more sophisticated concepts in the protocols of the most popular P2P systems there is still a gap between existing concepts and their adaptation to applications. Let‘s think of Bittorrent [Coh03], a recently emerged distributed P2P protocol. From a scientific view there is room for several improvements: Traffic overhead is to large due to the use of HTTP and internally hashing could be improved dramatically with using hash trees instead of lists. ISP's might get crazy when they think how much costs could be reduced with a switch to a more spare protocol at that particular user size. P2P developers can use frameworks like JXTA that offers abstraction from the network layer to build scalable P2P applications. Despite JXTA is originally written in Java there are several bindings to other languages like c and python.

3.2 Security

Security can be perceived as a double sided sword for P2P. With todays aggregration between tiny data-critical local area networks and huge autonomous dangerous networks security is one of the major for developers and users. Distributed implementations create additional challenges for security compared to client-server architectures. Conventional security mechanisms like firewalling and anti-virus software do not work for pure P2P environments at all or at least have to be adapted. While there are various security-related concepts for distributed systems [FM02, RFH$^+$01, DFM01] current file sharing systems don't offer too much security concepts. Many users seem to be afraid of file sharing applications because of the enormous risk that comes along with unchecked data. This led to a paradigm where P2P is associated with security risks applications. But on the other hand P2P technology does not only offer new security problems. Usually, centrally implemented solutions like anti-virus software and anti-spamfiltering can integrate P2P technology to be more powerful. Think of the traditional file sharing applications: Distributing data can not only be used to deploy viruses and worms on the end-users computers but can help to deploy anti-virus definitions in a discrete, secure space. Vipul's Razor [Raz] is an example for P2P spamfiltering. It uses a distributed, collaborative, spam detection and filtering network. Through user contribution, Razor establishes a distributed and constantly updating catalog of spam in propagation that is consulted by email clients to filter out known spam.

3.3 Metadata

Locating data like mp3-encoded music in a P2P system could be simply implemented with query broadcasting, though this is not quite efficient [RFI02] or with distributed hash tables with is more challenging but powerful. Answering more complex queries like *Give me textual information about Christoph Columbus before he discovered America* seems to be a very distinct, far more complex task. For example, traditional P2P networks are not even able to search for Mangas in a pool of video files, since avis and mpgs don't annotate something like a movie type. Mp3-Encoded Music files often comprehend ID3-tags with information like song title, artist, year of appearance and music style. None of the current popular file sharing systems support searching including this type of information due to ascending message traffic overhead. To convert such raw data into information or at least into correct content and therefore make it to locatable and traceable information appropriate data management concepts are necessary. Edutella [N01] is a prototype E-Learning P2P system. It based upon Metadata standards defined for the WWW and uses an RDF-based metadata infrastructure for P2P applications. Another application that adapts metadata structures is Lionshare[LI04].

Lionshare is an academic-oriented, P2P (P2P) networking technology that merges secure and expanded electronic file-exchange capabilities with information gathering tools into a single, open source application. In contrast to Edutella it uses an own appropriate metadata schema . Though both systems are deployed in small environments they show us where P2P can be successful beyond the scope of file sharing in next generation systems.

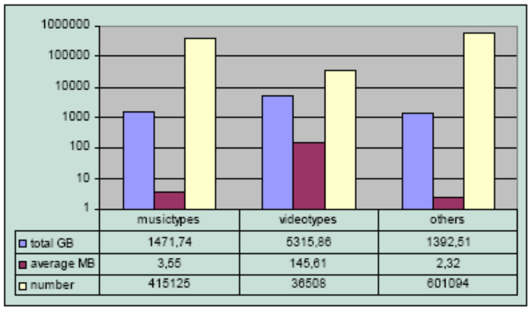
4 Trends

Besides the challenges for developers of innovative, new P2P applications we should take a look at todays most popular systems, even if this is pure file sharing again. These systems comprehend what more advanced prototype systems lack of and what makes P2P that famous, the huge amount of users. Furthermore we should look at the content distribution in such systems to get knowledge about the user behavior and which amount of which particular data types is apparently shared.

4.1 Content

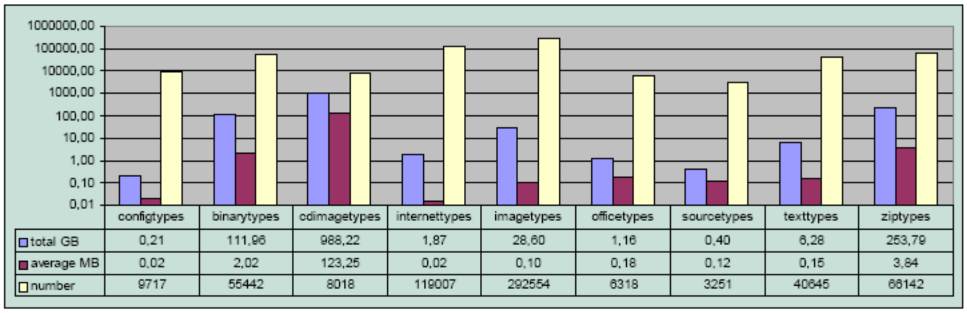
While the video industry tries to avoid the failures the music industry did since Napster emerged, its definitely worth to look into detail of those systems and find out if video files have become a factor. To collect the types and the amount of content we implemented an analyze tool that connects to randomly chosen hubs on the Direct Connect [Neo] P2P network. It collects data lists from connected users on those hubs and classifies types by using a huge table of commonly known file extensions.

Figures 1 depicts that music topic is most popular of all content but has lost its superior position when it comes to absolute data size to video content. Music is only top in the total number of files, however a music file is much smaller than a video file (3.6 MB vs. 145.6 MB) its total amount is significantly smaller (1.4 TB vs. 5.3 TB). Since the other class of Figure 1 comprises more files in total and by number than the popular classes together, we have a deeper look on the other classes. The results are depicted in Figure 2. This study proofs that (apart from popular video and music content) users are willing to share huge amounts of various data like text data, source files, web pages and images. That content outnumbers music and video in terms of total number but is less in total space then music or video files. ISP's mainly have to be concerned with the total size of certain contents. Due to the traffic they generate they still can keep their view on file sharing systems as multimedia sharing. However for P2P developers this trend seems to be highly interesting and Next-Generation P2P Systems. Lets think



	musictypes	videotypes	others
▢ total GB	1471,74	5315,86	1392,51
■ average MB	3,55	145,61	2,32
□ number	415125	36508	601094

Figure 1. Popular classes of data on Direct Connect Hubs (log scale)



	configtypes	binarytypes	cdimagetypes	internettypes	imagetypes	officetypes	sourcetypes	texttypes	ziptypes
▢ total GB	0,21	111,96	988,22	1,87	28,60	1,16	0,40	6,28	253,79
■ average MB	0,02	2,02	123,25	0,02	0,10	0,18	0,12	0,15	3,84
□ number	9717	55442	8018	119007	292554	6318	3251	40645	68142

Figure 2. Collection of other data beside music and video on Direct Connect Hubs (log scale)

of on integrated browser or P2P collaboration tools for coding when we look at the vast amount of html- and programming files we find in this study.

4.2 Popular systems

Cachelogic recently published a survey about P2P traffic to advertise for a new network analysis tool called Streamsight [Cac]. They measured network traffic at leading ISP's over a six-month period to get more advanced statistics than any other study about P2P networks before. The most obvious result was that P2P applications outweighed all other web traffic by far. While industry and media has published several announcements in the last month that P2P traffic is on decline due to their attacking on the platforms or on single users. Cachelogic states that P2P traffic is mostly stable if not rising. Since more and more P2P applications use dynamic or variable ports or mask the traffic traditional simplified TCP measurements do not work anymore. Even known ports like 25 or 80 could not be classified to be truly web and mail since P2P applications even use those ports, e.g. JXTA uses HTTP to get around routers and firewalls. So as to classify traffic this way its necessary to look into packages more detailed that just to watch ports. When it comes to file sharing popularity media still focuses on famous platforms like Edonkey and Kazaa or applications that use famous protocols like

Gnutella and Fasttrack. Cachelogic states that Kazaa is on decline and Edonkey is on rise but are only second and third compared to the definitely leader when it comes to creating traffic for ISP's, Bittorrent [Coh03]. Though Bittorrent as a distributed protocol quite like ftp is more a service then a platform, it has become the dominating P2P system in a short time. There are several clients like Bittornado and ABC which basically implement the standard client with a few modifications . Since Bittorrent is such a huge P2P traffic creator it outnumbers HTTP traffic easily and for ISP's is the number one to deal with. Again its mostly interesting to see how such an simplified protocol with huge set of improvements was able to get such an attention in a short time while scientists develop several more sophisticated concepts which are not being adapted to such system because of their complexity.

5 Summary

This paper briefly summarized current trends and upcoming challenges in the P2P area from a scientific and application-driven view. While file sharing applications still remain the most popular P2P examples file sharing users changed platforms and more interesting their data. Furthermore today there are several solutions to problems like scalability and security as there are promising concepts like metadata to take P2P to the next level.

References

[Cac] Cachelogic. Cachelogic 2004. http://www.cachelogic.com.

[Coh03] Bram Cohen. Incentives build robustness in bittorrent, 2003.

[CSWH01] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A distributed anonymous information storage and retrieval system. *Lecture Notes in Computer Science*, 2009:46+, 2001.

[DBK+01] Frank Dabek, Emma Brunskill, M. Frans Kaashoek, David Karger, Robert Morris, Ion Stoica, and Hari Balakrishnan. Building peer-to-peer systems with Chord, a distributed lookup service. In *Proceedings of the 8th Workshop on Hot Topics in Operating Systems (HotOS-VIII)*, pages 81–86, 2001.

[DFM01] Roger Dingledine, Michael J. Freedman, and David Molnar. The free haven project: Distributed anonymous storage service. *Lecture Notes in Computer Science*, 2009:67–95, 2001.

[FM02] Michael J. Freedman and Robert Morris. Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002)*, Washington, D.C., November 2002.

[Gnu] Gnutella2.com. Gnutella. http://www.gnutella2.com.

[KBC$^+$00] John Kubiatowicz, David Bindel, Yan Chen, Patrick Eaton, Dennis Geels, Ramakrishna Gummadi, Sean Rhea, Hakim Weatherspoon, Westly Weimer, Christopher Wells, and Ben Zhao. Oceanstore: An architecture for global-scale persistent storage. In *Proceedings of ACM ASPLOS*. ACM, November 2000.

[KL04] M. Kolweyh and U. Lechner. Data mining in peer-to-peer systemen. In *Virtuelle Organisation und Neue Medien 2004, Workshop GeNeMe2004 Gemeinschaften in Neuen Medien*, pages 103–114, Josef Eul Verlag, Lohmar - Köln, 2004. M. Engelien and K. Meißner.

[LRW03] N. Leibowitz, M. Ripeanu, and A. Wierzbicki. Deconstructing the kazaa network. In *3rd IEEE Workshop on Internet Applications (WIAPP'03)*, Santa Clara, CA., 2003.

[MKL$^+$02] Dejan S. Milojicic, Vana Kalogeraki, Rajan Lukose, Kiran Nagaraja, Jim Pruyne, Bruno Richard, Sami Rollins, and Zhichen Xu. Peer-to-peer computing. Technical Report 57, HP Labs, 2002.

[Mon01] Alberto Montresor. Anthill: a framework for the design and analysis of peer-to-peer systems. In *4th European Research Seminar on Advances in Distributed Systems*, Bertinoro, Italy, May 2001.

[Nap] Napster Inc. Napster.com. http://www.napster.com.

[Neo] NeoModus. Direct connect. http://www.neo-modus.com.

[Raz] Project: Vipul's Razor. Vipul's razor. http://razor.sourceforge.net.

[RFH$^+$01] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. A scalable content addressable network. In *Proceedings of ACM SIGCOMM 2001*, 2001.

[RFI02] M. Ripeanu, I. Foster, and A. Iamnitchi. Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. *IEEE Internet Computing Journal*, 6(1), 2002.

[SMK$^+$01] Ion Stoica, Robert Morris, David Karger, M. Francs Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the 2001 conference on applications, technologies, architectures, and protocols for computer communications*, pages 149–160. ACM Press, 2001.