Analysis and characterization of Peer-to-Peer Filesharing Networks

J. LLORET MAURI[1], G. FUSTER[2], J. R. DIAZ SANTOS[1], M. ESTEVE DOMINGO[1]
[1] Department of Communications, [2] Illes Balears d'Innovaciò i Tecnologia
[1] Polytechnic University of Valencia [2] Edifici 17, Parc Bit, Cra. Valldemossa
[1] Camino Vera s/n, 46022, Valencia [2] 07021, Palma de Mallorca
SPAIN

Abstract: Since Peer-To-Peer file-sharing networks appearance a few years ago, many Internet users have chosen this technology to search for programs, films, songs, documents, etc. This number of users is growing every day. The main reason has been the content (in occasions illegal) that can be found and downloaded over these networks. This article deals with the analysis and characterization of eight P2P Public networks: *Gnutella, FastTrack, Freeenet, BitTorrent, Opennap, Edonkey, Soulseek and MP2P*. Finally, the authors will show a relationship between their characteristics and, in six of them, between their number of users, files shared and the amount of data shared in their networks

Key-Words: - P2P, Analysis P2P, Filesharing Networks, Overlay Networks

1 Introduction

The number of users connected to public P2P Networks is increasing day by day. Actually, there are a great variety of P2P networks and some of them with a lot of P2P clients. One of the first steps is to differenciate between P2P network and P2P clients. P2P networks are a set of rules and interactions that allow P2P clients to communicate. A P2P client is a computer application that allows a user interact with other users in the same network. The number of P2P emergent networks is continuously increasing and their clients are having more and more capabilities every time.

P2P filesharing is one of the Peer-To-Peer variants that is accummulating more and more participants. Although, there are users that try to download files from the network, without intention of providing any, there are a lot of users who are able to share what they have with the whole community without caring about who is downloading their files.

The success of a P2P network inside a user community is determined by several factors:
- Simplicity: a P2P network with a graphical and easy-to-use P2P client is always welcome
- Language: a P2P client with Multilanguage support allows a broader deployment amongst international users.
- Download speed: some P2P networks, due to their internal behavior, are optimal for downloading files of reduced size. Others, however, use multisplitting mechanisms and permit the download from multiple sources, making them suitable for obtaining larger files.

These parameters are responsible for a the increasing popularity of some networks, whilst others are disappearing. These factors can make a P2P network becoming more attractive to users of a specific nation due to the utilization of a concrete language or even social trends [1]. If a P2P client changes its P2P network, all its community users will remain using it. As an example, many users have remained 'loyal' to the Morpheus P2P client throughout its evolution [2].

The P2P overlay network protocols are located in the application layer. These protocols can be programmed to run over TCP or UDP; however, it is possible to use new ones like DTCP that works directly over IP [3].

The communication between the clients of a network, the transferred data and the routed data are done independently of the lower layers of the communication protocol stack.

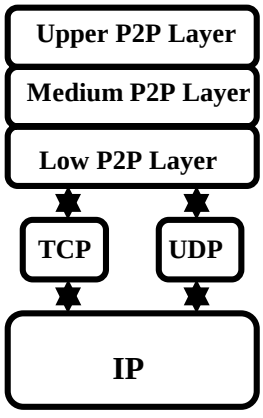

Figure 1. Three Peer-To-Peer sublayer model

A lot of P2P network protocols divide the P2P layer into a model of three sublayers (see Figure 1):
- Low layer: Responsible for communication, user authentication, network discovery, etc.
- Medium layer: Data search, file exchange, management, data routing, etc.
- Upper layer: Applications such as instant messaging, storage systems, processing systems, etc.
Some features, such as security, must be addressed at all three layers.

2 Motivation

Currently there are a lot of P2P filesharing networks in existence , and many of them have millions of on-line users. The main public Internet P2P filesharing networks are Gnutella [4], FastTrack [5], Freenet [6], BitTorrent [7], Opennap [8], Edonkey [9], Soulseek [10] and MP2P [11], although there are other networks that are not so popular. [12]. We have selected the eight most popular networks due to their different type of working architecture in order to analyze their features and classify them.

What a user really wants is to find the file that he is looking for. But this file is not always in the network where the user is searching. On the other hand, there is a big probability to find, for example, an audio file if it is being searched in a network where only audio files are shared. Most of the networks implemented nowadays support any filetype.

There are some actual P2P software clients that are able to use more than one P2P protocol and they can join several networks. Some of them are Shareaza [13], MLDonkey [14], Morpheus [15] and cP2Pc [16]. However, the use of this solution, in order to search a file, means that the user has to be permanently connected to all networks. On the other hand, if a client is developed that is able to join all networks, the computer running this client will need a lot of processing capacity, and , if a new P2P filesharing network is developed, a new client is required to support the new architecture and all users will have to update their client to join the new network.

What is needed is a system which will allow to search in every P2P network and download from every peer of every network. To do so, the architectures mentioned above are analyzed, classified and its users, files shared and amount of data shared measured, in order to find the best way to interconnect them in future works.

3 P2P architecture analysis

First of all it is required to know which are the coomon features in the P2P networs beign analyzed , and which are different.

A lot of P2P filesharing architectures have the following common features [17]: user privacy, encryption, distribution, data redundancy, direct transfer and high availability.

There are several parameters that can be changed in these architectures: decentralization, routing algorithms and metrics, load balancing, traffic balancing data search motor and file downloading system.

3.1 Kind of architecture

Based on their architecture, P2P networks can be decentralized, centralized or partially centralized. P2P software applications communicate between them in order to exchange data. These applications allow a peer to become a server and a client at the same time, these peers are called servents. In the decentralized networks, no element of the P2P network is essential for the system to operate; otherwise, in the partially centralized or centralized networks there are some elements with a bigger status and they are necessary for the system to function. In both cases, the data transfer is made directly between the edge clients, without any central server as an mediating of this transfer.

3.1.1. P2P Decentralized architectures.

In decentralized P2P networks all computers have the same responsibility and capacity. Therefore, a certain node can make data requests to other nodes and, at the same time, solve and answer the requests from other ones. In this architecture, nodes can play three roles: as a server when it is asked for data from a node, as a client, when it asks for data to another node and as a router, when the node is passing data between other two clients. A node employs several algorithms to make searches, for example, using a list of known nodes or sending a multicast or broadcast message to the network. In the pure P2P architecture there are three basic actions: search of active nodes, enquiry of resources and the content transfer. In the search, the node sends broadcast or multicast ping messages to the network. The active nodes will answer with a pong message. After that, the node will send a query, which will be replied by those nodes with the requested resource. Later, the user will be able to select the resources that he wishes to download. The decentralized P2P analyzed networks are Gnutella, and Freenet. In this kind of architecture we can locate other not analyzed networks in this article such as CAN[18], Chord [19], Pastry [20] and Tapestry [21].

3.1.2. P2P Centralized architectures.

In a centralized P2P architecture a central server is used and not all the nodes have the same performance and the same functions. These architectures can be considered as P2P systems since the nodes communicate between themselves directly. Two types of centralized P2P architectures can be differentiated: the one where nodes consult services and the ones where nodes and resources consult services. In this paper, only the second one will be considered, due to the fact that it is the one used in P2P filesharing networks. In this kind of centralized P2P networks, a central server has the role of storing the active nodes and the indexes of shared contents. There are three basic actions: register action, consult action and content transfer action. During the register action, a P2P client will inform to the server that it is active and the contents it has for sharing. When a request is performed, it will send the information about the desired file for downloading. When the server receives a query it will be processed in two ways: by the search in indexes or by sending the consultation to connected nodes. Then, the reply will be sent back to the original node so the client can select the resources to be transferred to download to his computer. The centralized architecture with node and resources query service analyzed in this article is Soulseek.

3.1.3. P2P Partially centralized architectures.

There are two types of partially centralized architectures. The first ones are similar to centralized architectures, but instead of a single server there is a farm of servers with a P2P network at this level. The second ones are similar to decentralized architectures, but there are some nodes called *supernodes* that act as a central node. This *supernodes* will perform the search for other supernodes in order ot find the requested file. The partially centralized architectures analyzed here are BitTorrent, OpenNap, Edonkey and MP2P in the first case and FastTrack and Gnutella 2 in the second case.

3.2. Discovery and search algorithms

In order to find a file in a P2P network, a search is needed. The implemented search algorithm in every network depends on the kind of the network (centralized P2P, decentralized P2P, and so on). There are several types of algorithms [22] and they are covered below:

3.2.1. Centralized indexes and repositories Model (CIRM):

In this model P2P clients are connected to a central server where they publish their shared files and some data such as the name, the size, etc. The central server keeps a database with the indexes of the clients and their contents and it allows to do searches about it. After a search it will send back the name of the files that match with the search, together with reference data and index of the client or clients having it. This model is used by the Soulseek network.

3.2.2. Distributed Indexes and Repositories Model (DIRM).

In this model there is a group of available servers called "brokers". Each "broker" has the indexes of the local clients and in some cases the indexes of some files from neighbour "brokers". When a client performs a query to a "broker", this one searches in its local database and if it doesn't find a match, it uses the local index in order to find a neighbour "broker" that can send the request. The server indexes are not static and can change according to the files in the system. The networks OpenNap, eDonkey, MP2P and BitTorrent use this model.

3.2.3. Flooded Queries Model (FQM).

The P2P clients in this model perform queries to all of their directly connected neighbours (broadcast). If the neighbour has the content, it replies. Otherwise if floods the query to its neighbours. This model is used by the Gnutella network.

3.2.4. Selective Queries Model (SQM).

This model is based in the model of flooded queries, but in this case the requests are sent to specific clients which are considered to have the greater probability finding the request. The clients with a higher bandwidth and process capacity will be considered automatically "superpeers". Those clients with less bandwidth will be "superpeers" clients. This type of system uses an flow control algorithm for sending queries and replies. It also has a diagram of priorities used to discard some messages. This model is used by FastTrack and Gnutella 2 [23].

3.2.5. Documents Routing Model (DRM).

This model is based in Distributes Hash Tables (DHT), where the data is placed in numerous nodes. In order to publish a document, it is routed to the client whose ID is the most similar to the document's ID. The process is repeated until a close match is found. This model is used by Freenet , CAN, Chord, Pastry and Tapestry.

The analyzed architectures and discovery and search algorithms organization can be seen in Figure 2.

3.3. File downloading system

In the case of an interrupted download, modern P2P networks allow for it to resume (file resume). Different download systems exist and they are covered below.

3.3.1. Single-source download.

The file is downloaded from one or several sources, but not simultaneously. This download type is used by Freenet, Soulseek and MP2P.

3.3.2. Multi-source download.

The file (or parts of it) is downloaded from multiple sources allowing a much faster download.

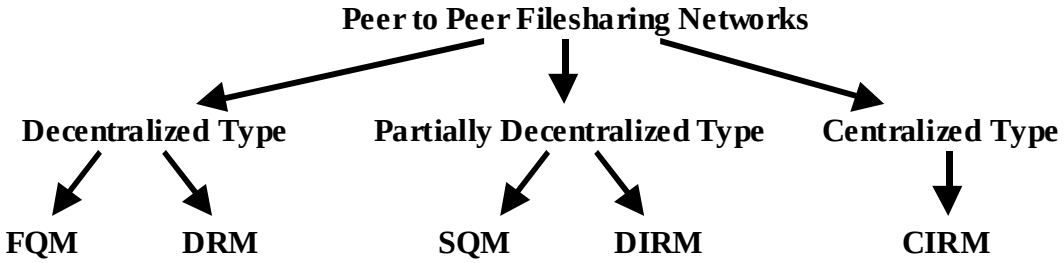

Figure 2. Analyzed architectures and discovery and search algorithms organization.

	Advantages	**Disadvantages**
Decentralized P2P network	No single point of failure	Slow discovery.
Partially Centralized P2P network	Fast Searches	Several points of failure Wide geographical dispersion
Centralized P2P Network	Fastest Searches	Single point of failure Single index repository Obsolete search results
The centralized indices and repositories model	Short query time Simple	Not scalable Server failure disables queries Large support needed at server
The distributed indexes and repositories model	Tolerant to server failures.	Need to avoid obsolete updates between "brokers".
The flooded queries model	Efficient for small communities	Limit request TTL (Time to Live).
The selective queries model	Larger bandwidth and scalability	It can be exposed by an intermediate intrusion
The documents routing model	Quick search The system is scalable	Any client in the network can change the results of the system
Multi-source downloads	Fast download of small files	Use segmented multi-source download for larger files

Table 1: Advantages and disadvantages of the analyzed items

Normally a "hash" is assigned to the files that allow the client application to complete and verify the download.. The downloading process is performed from the beginning of the file to the end of it. This type of downloading system is used by Gnutella, FastTrack, OpenNap.

3.3.3. Segmented multi-source download.

It is similar to the previous one, but it allows downloading parts of the file that are not sequential. In order to perform this type of downloading, the client application segments the file in metadata. BitTorrent and eDonkey use this type of download.

3.4. Advantages and disadvantages

Each architecture, each model and each file downloading system is the best according its situation. Table 1 tries to sum up the advantages and disadvantages of the items 3.1, 3.2 and y 3.3.

3.5. Users, Number of files shared and amount of shared data.

Some ISPs have observed that their networks became rapidly congested and sometimes P2P traffic reached about 60% of the total traffic [24]. Although not so striking, Internet2 administrators also computed impressive results on 16 February 2004 where 10.46% of the total traffic was originated by P2P file-sharing [25]. CAIDA (Cooperative Association for Internet Data Analysis) also shows that Internet traffic is mainly dominated by P2P file-sharing protocols and HTTP [26]. Some articles show the average number of connected users in some architectures [27] and sometimes even the maximum number of users. Some papers even study the economic cost of downloading a file analyzing the required time [28]. The manner in which the number of connected users is calculated is sometimes deceptive if it is solely based on the amount of users that download a certain client program for a P2P architecture [29].

In order to measure P2P parameters, we have taken one totally decentralized architecture (Gnutella [4]), four partially decentralized architectures (FastTrack [5], OpenNap [8] eDonkey [9] and MP2P [11]) and a centralized architecture (SoulSeek [10]). For the purpose of this article we took measurements between November 2003 and February 2004. To take

measurements of the corresponding architectures, the most adequate clients have been selected, bearing in mind those that would provide the most information on the architecture or the highest update frequency to measure the parameters. The Gnutella architecture has been analyzed with the Limewire client, but taking into account the statistics taken by Limewire's web page [31]. This is the reason why the number of users, files and size of shared data is relatively low. In the FastTrack architecture, the measurements have been taken with the KaZaA Lite client. In order to analyze OpenNap architecture, the Napigator client has been used. The eMule client has been utilized to analyze the Edonkey architecture. The MP2P architecture has been analyzed by means of the Blubster client. Finally the Nicotine client has been used to analyze the SoulSeek architecture. The measurements taken have been compared with data obtained a year ago. This shows that older networks are decreasing (Gnutella, FastTrack and OpenNap), due to the creation of new P2P networks (eDonkey, MP2P and Soulseek) that attract users from older ones. The total number of users connecting to the P2P file-sharing networks is growing. Therefore, the number of users increasing Internet traffic due to the use of these networks is growing.

Figures 3, 4 and 5, show the average measures taken for users, number of shared files and size of data shared.

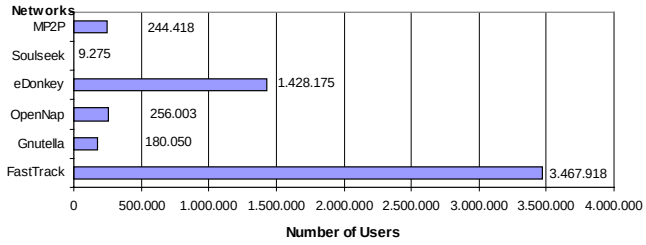

Figure 3: Users in public P2P Networks

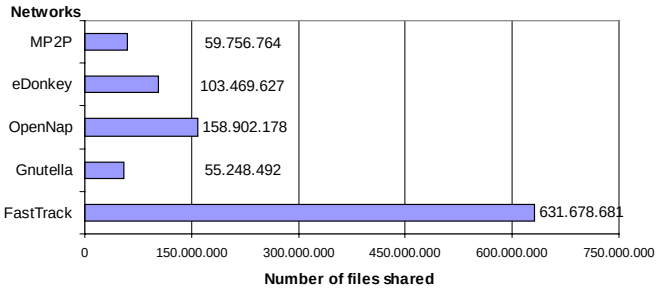

Figure 4: Number of files in Public P2P Networks

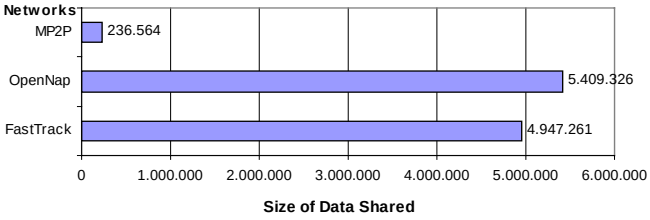

Figure 5: Total Size of Data Shared in Public P2P Networks

In some cases the variation of its users, files shared and amount of data shared, along a week, could be over ±50%.

The hours where all the architectures measured, except MP2P, have more users are between 18:00 and 1:00 according to the GMT+01:00 timezone.

3.6. Analyzed architectures summary

The table 1 tries to sum up all previous analyses in a comparative way to let us obtain a global perspective of the analysis taken. In Gnutella and eDonkey networks, their clients let us measure the amount of data shared. In Soulseek network the clients let us know the users in this network only. Due to the architecture of Freenet and BitTorrent, the average number of users, files shared or total size of files shared in these architectures can not be taken.

	P2P Architecture	Discovery and Search Algorithm	File Download System	Files type	Protocol	# of users	# of shared files	Amount of shared data (in GB)
Gnutella	Decentralized	FQM/SQM *	Multisource	All	TCP	180.050	55.248.492	n/t
FastTrack	Partially centralized	SQM	Multisource	All	TCP	3.467.918	631.678.681	4.947.261
Freenet	Decentralized	DRM	Single-source	All	TCP	n/t	n/t	n/t
BitTorrent	Partially centralized	DIRM	Segmented Multisource	All	TCP	n/t	n/t	n/t
OpenNap	Partially centralized	DIRM	Multisource	All	TCP	256.003	158.902.178	5.409.326
SoulSeek	Centralized	CIRM	Single-source	All	TCP	8981	n/t	n/t
eDonkey	Partially centraliced	DIRM	Segmented Multisource	All	TCP, UDP	1.428.175	103.469.627	n/t
MP2P	Partially centraliced	DIRM	Single-source	Audio	UDP	244.418	59.756.764	236.564

Table 2. Analyzed architectures comparative (* SQM in case of Gnutella 2, n/t: measure not taken)

4 Conclusion

Eight working environments with a considerable number of users have been analyzed, considering the kind of the architecture, the search algorithm, the common parameters, the transport layer protocol and the downloading system in each of them.

All these networks have their advantages and disadvantages and each of them performs better than the other ones according to the environment where it is implemented or according to a desirable parameter.

All analyzed networks are unstructured Peer to Peer networks.

The graphs of users, files or size of total files shared do not depend on the decentralization degree of the architecture. There could be more users in an architecture (for example eDonkey) than in others (for example OpenNap); otherwise, there are more files shared in OpenNap than in eDonkey. Total size of shared data does not depend on the number of files shared in the architecture. FastTrack architecture is the one which has the most files, otherwise, the one which has the most total size of shared data is OpenNap. The OpenNap and MP2P architecture have practically the same number of on-line users; however, there are three times more shared files in the OpenNap network than in the MP2P architecture.

Observing the obtained graphs we can establish a certain relationship, it is more probable to obtain the desired content in networks with more users connected.

References:

[1] Regional Characteristics of P2P: File sharing as a multi-application, multi-national phenomenon, Sanvine White Paper, October 2003

[2] KaZaA Times http://www.kazaa.times.lv/telo_p2p_fasttrack.htm

[3] Alpine http://peertech.org/alpine/overview.html

[4] Eytan Adar and Bernardo Huberman. Free riding on gnutella. First Monday, 5(10), October 2000.

[5] Nathaniel Leibowitz, Matei Ripeanu, and Adam Wierzbicki. Deconstructing the Kazaa Network, , 3rd IEEE Workshop on Internet Applications (WIAPP'03), June 2003, San Jose, CA

[6] I. Clarke et al. Freenet: A distributed anonymous information storage and retrieval system, ICSI Workshop on Design Issues in Anonymity and Unobservability, Int'l Computer Science Inst., 2000.

[7] Bram Cohen. Incentives Build Robustness in BitTorrent, Workshop on Economics of Peer-To-Peer Systems Berkeley CA June 2003

[8] OpenNap http://opennap.sourceforge.net/

[9] Oliver Heckmann and Axel Bock. The eDonkey 2000 Protocol. Technical Report KOM-TR-08-2002, Multim. Communications Lab, Darmstadt University of Technology, December 2002.

[10] Soulseek http://www.slsk.org

[11] MP2P http://www.blubster.com/protocol1.html

[12] Wikipedia http://www.wikipedia.org/wiki/Peer-to-peer

[13] Shareaza http://www.shareaza.com

[14] MLDonkey http://mldonkey.berlios.de/

[15] Morpheus http://www.morpheus.com

[16] cP2Pc: Integrating P2P networks. Ihor Kuz, Maarten van Steen, www.nlnet.nl/project/cp2pc/20030620-cp2pc.pdf

[17] J. Walkerdine, L. Melville, I Sommerville, Dependability Properties of P2P architectures, Second International Conference on Peer to Peer Computing, 2002.

[18] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker, A Scalable Content-Adressable Network, ACM Sigcomm 2001

[19] I. Stoica, R. Morris, D.Karger, F.Kaashoek, H. Balakrishnan, Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications, ACM Sigcomm 2001

[20] A. Rowstron and P. Druschel, Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems, IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), heidelberg, Germany, pages 329-350, November, 2001

[21] B. Zhou, D.A. Joseph, J. Kubiatowicz, Tapestry: a fault tolerant wide area network infraestructure, UC Berkeley technical report UCB/CSD-01-1141

[22] TCD 4BA2 Project http://ntrg.cs.tcd.ie/undergrad/4ba2.02-03/p8.html

[23] Gnutella2 http://www.gnutella2.com

[24] Peer-to-Peer File Sharing: The impact of filesharing on service provider networks, Sandvine Incorporated, 2002

[25] Internet2 NetFlow, Weekly reports, info at http://netflow.internet2.edu/weekly/20040216/

[26] The CAIDA website at http://www.caida.org

[27] Stefan Saroiu, P. Krishna Gummadi and Steven D. Gribble, A Measurement Study of Peer-to-Peer File Sharing Systems, Department of Computer Science & Engineering, Univ. of Washington, Tech Report UW-CSE-01-06-02.

[28] Artur Marques, avaliable at http://arturmarques.com/docs/economics/arturmarques_dot_com_freeloading.pdf

[29] Corporate P2P Usage and Risk Analysis, AssetMetrix Research Labs, July 2003

[30] Statistics in Limewire.com http://www.limewire.com/english/content/netsize.shtml