Networking Laboratory
Department of Electrical and Communications Engineering
Helsinki University of Technology

Peer to Peer and SPAM in the Internet

Report based a Licentiate Seminar on Networking Technology Fall 2003

Editor: Raimo Kantola

Abstract:

This report is a collection of papers prepared by PhD students on Peer-to-Peer applications and unsolicited e-mail or spam. The phenomena are covered from different angles including the main algorithms, protocols and application programs, the content, the operator view, legal issues, the economic aspects and the user point of view. Most papers are based on literature review including the latest sources on the web, some contain limited simulations and a few introduce new ideas on aspects of the phenomena under scrutiny. Before presenting the papers we will try to give some economic and social background to the phenomena. Overall, the selection of papers provides a good overview of the two phenomena providing light into what is happening in the Internet today. Finally, we provide a short discussion on where the development is taking us and how we should react to these new phenomena.

Acknowledgements

My interest in Peer-to-peer traffic and applications originates in the discussions that took place in the Special Interest Group on Broadband networks hosted within the TEKES NETS program during 2003. In particular, I would like to thank Pete Helenius of Rommon for his insights. The Broadband group organised an open seminar in Dipoli on November 20th that brought the topic into a wider discussion involving people from both the networking side as well as the content business.

I would also like to acknowledge the inspiring discussions with Juha Oinonen and Klaus Lindberg of CSC and Funet on the subject of peer-to-peer traffic and applications.

Preface

This report is based on the work done by my licentiate and Ph.D students on the Licentiate Course on Networking Technology (S38.030) during the Fall 2003. The idea was to learn about the disruptive and also annoying phenomena that have become very commonplace over the past couple of years in the Internet: namely, the Peer-to-Peer traffic and applications and the unsolicited and unwanted e-mail or Spam. Due to the illegal copies of audio and video files in Peer-to-Peer networks and unwanted nature of Spam one could claim that both of these phenomena are parasitic and threaten the purpose the Internet was designed for. Especially, the proponents of p2p say that Internet was originally all peer to peer and it was only later when the client server approach took hold and became widespread.

The nature of information goods on the Internet is such that it is very difficult to earn money on them. This applies equally to content and services. This has brought the advertisement driven business model to the Internet. Internet charging does not much care about the direction of the traffic, nor usually for the volume. Under these conditions sending of bulk unwanted, unsolicited, commercial or non-commercial e-mail or spam has emerged. Spam is a clear violation of the original cooperative nature of the Internet.

The students were given assignments. They prepared a paper and presented it during two seminar days late November 2003. The presentations were opponeered and discussed in the seminar and the students had the opportunity to improve their papers. The best of the student papers are included in this report. I have used some time to polish the papers to make them more readable and also given comments to a few of them as Editor's notes that appear in the Footnotes. Nevertheless, I apologize for the remaining bugs in the text as well as defects in the layout. We preferred timeliness of publication and left final polishing of the text to future work on those papers that warrant it.

The idea of the seminar was to learn about the new phenomena and try to understand where these phenomena are leading the Internet in the near future. I hope this report will pass on what we learned and give some material for thought of your own.

January, 2004 Raimo Kantola

Disclaimer:
Some of the students mention their Affiliations. The reader should be aware that the students are responsible for the contents of their papers and nothing in them represents an agreed official position of the Affiliation concerned. Nor does anything said in this Report represent any official legal position of the Networking Laboratory.

Table of Contents

Part I: Peer-to-Peer applications:

Introduction

The introduction of broadband to the residential market in the Internet has brought always- on connectivity to the wide public. Broadband makes possible many new services. Transporting audio and video on the NET is now possible. The World Wide Web has moved towards a TV –like user experience with a rather clear separation of the roles of content providers and consumers. The users are content producers only to a rather limited extent. From the user's point of view, he or she has invested in a powerful PC with lots of memory and cycles that are only lightly loaded most of the time. The user is also paying a flat rate monthly charge for the connectivity to the Internet irrespective of the volume of traffic the user is sending or receiving. This is a natural environment for the new popularity and the emergence of new Peer-to-Peer Applications. The idea is to use the PCs of the users to do something useful cooperatively without relying on any service provider's help for which the users would have to pay. Most popular use of the p2p applications falls under the category of File Sharing. Most of the files seem to be audio and video and a lot of that is originally illegal copies. Other, either clearly useful, benign or useful but disruptive to the existing value chain uses of the p2p technology have also emerged. The share of p2p traffic in all the Internet traffic has grown dramatically over the part 2…3 years. It is now said to anything between 20 and 80% of all Internet traffic.

All in all it seems that only recently many were asking why is broadband needed and what is the killer application in broadband? Now that, the users have surprised the incumbent players again and found the killer application, many people are unhappy. The content owner lobby used to protecting its business that is based on copyrights has launched a full-scale attack on peer-to-peer applications and their users.

The new popularity of Peer-to-peer started with Napster that still had a central server for storing index information to files that the users were willing to share. Only the sharing of files itself took place in a peer-to-peer fashion directly from host to host. The central server in Napster turned out to be vulnerable to a legal attack of the copyright lobby and Napster was closed down. Recently, Napster has been reopened as a commercial service. The commercial Napster, however, has very little to do with the original Napster in terms of implementation.

Very quickly after the demise of Napster the scene was taken over by fully decentralized implementations of File Sharing applications such as Gnutella, KaZaA, DirectConnect etc. A decentralized network of peers has no central authority and nodes can freely join the network at any time and disconnect from the network without loss of operability of the whole system.

It is easy to fall into the trap of taking a moral stand and say that Peer-to-peer File Sharing is evil, it is stealing from the authors of content and it should be punished severely. I believe, it is important to keep a cool head and try to understand the phenomena from different angles. One important angle is economic. In a couple of next sections I will briefly analyse the economics of information in a networked society.

Information value chain

In the digital information economy the simplest value chain makes the distinction between content providers and the distribution chain. Under content providers we lump the audio recording and movie industry as well as newspapers and other publishers. Also the distribution channel is diverse. It contains the traditional methods used by each of the content provider types. Now, with the emergence of the broadband Internet, a more efficient method of digital content distribution has come to being. Besides the Internet many other components are needed; those however, are sold as consumer goods directly to the users of the network with consumer market economic rules.

The content industry is used to earn money based on its copyright. The nature of copyright is that it gives a monopoly right to the copyright owner to earn money on the content for tens of years. It is a legal mechnism imposed on top of the markets. Monopoly is adverse to efficiency and not surprisingly in this case the copyright industries have been slow to embrace the

Internet as a distribution channel. The more efficient distribution channel is trying to push itself onto the market fighting the monopoly driven content distribution. Since the existing players are not using the new technology, the technology adoption takes place in an ad hoc and uncontrolled fashion. The players on the new scene are small companies and the users themselves.

As a result, we are a observing a major conflict of interests in the information economy. On one side we have the content owner lobby and on the other we have the businesses that are building and running the networks. One can claim that currently the most natural value chain in the information economy from content providers to content distributors and to users is broken. The creation of new broadband technology is relentless, it will continue and the conflict of interests is likely to see new forms. It seems obvious that sometime in the future a new balance of interests has to be found between the two ends of the value chain to mutual benefit.

Competition in the Information Economy

The Five forces model of competition in an industry proposed by Porter is widely used to model the dynamics of an industry. Let us use it for the information economy in general. At the center is information as goods in the most general sense. Figure A shows the wrappings around information goods and the five forces.

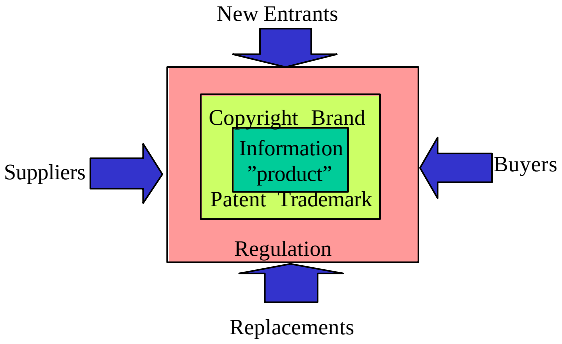


Figure A: Dynamics of Competition in Information Economy

Arrows in the Figure show the external forces. The fifth force, namely, the competition inside the industry is conditioned by the wrappings that are showns in the Figure.

The inner wrapping around the information good is formed mainly by the legal mechanisms that help the creator of the information good to earn money on the goods. The outer wrapping is the competitive regulation creating a maket like environment for the information industries.

The reason why the inner wrapping is necessary from an economic point of view is that information as goods is non-depletable. A non-depletable good is one, which can be given or sold to a buyer and retained by the seller at the same time. Under the condition of efficient digital technology including broadband networks the marginal copying costs of information are very low and approach to zero. This means that if no inner wrapping exists, the supply of information is infinite. It follows from the law of supply and demand that the price of information good under free competition also approaches zero. As a result, the free market alone is not able to create a price that would allow the information creator to earn a return on his investment and without means outside the market mechanisms all information economy would be unprofitable and no economy at all. The only remaining means to finance the creation of information would be taxation.

I see this dilemma as a fundamental lasting feature of information economy. The dilemma of the regulator is how to create market like conditions for the industries that have been given monopoly rights (copyrights and patents) by law.

There are two other important ways to earn money on information goods. One is the foundation of the wealth of, for instance, Microsoft, i.e. having a secret that everybody wants. It has proven to be a great money earner consistently producing margins that are well above average in the traditional product based industries. The last significant means of earning money on information is embedding it into a physical product. Although, it may be that the real value lies in the embedded information, the product can be sold on conditions of traditional product industries because the information may have little value without the wrapping of the the physical product surrounding the information. Another condition must be true as well: it must be difficult and expensive to make copies of the physical wrapping. An example of an industry that uses this method successfully is the cellular phone manufacturing starting with Nokia.

We should note that one can not freely choose the means of earning money irrespective of the type of information. An example is the audio recording industry that that uses for instance CDs as the physical wrapping for recordings. Since CDs are easy to copy, piracy is widespread. Piracy is widespread also in the production of expensive physical goods where a large portion of value is in the brand label (an information component of the product). It also must be relatively easy to produce the copies.

How does this relate to p2p? Peer-to-peer technology can be seen as a reflection of the market forces entering a monopoly driven industry. It also reflects a shift from creating new value by creating new information to creating value by using information in traditional industries. We will see in the next few years what the response of the traditional information industries will be, to what extent they will stuck to their copyright guns and to what extent they will create new business models that work even in the harsh environment of Broadband networks.

Scope of the papers

The student papers on Peer-to-Peer study a diversity of peer-to-peer applications, some fundamental concepts that lay behind the peer-to-peer technology, the impact of peer-to-peer onto the Internet Service Provider and the economic and legal aspects of peer-to-peer. A paper discusses secure computing that could be one way of protecting the rights of the content owners in the digital economy.

The papers are:

1. Gnutella project overview	Marina Shalomova
2. The Freenet project	Renjish Kaleelazhicathu
3. Skype - the p2p telephony overview	Markus Isomäki
4. Protocols for resource management in p2p systems	Qiu Yang
5. Performance measurements and availability problems in p2p	Johanna Antila
6. Modeling of Content location	Bai Xiole
7. Peer to peer traffic and ISP	Aki Anttila
8. Peer to peer file sharing and content distribution systems	Evgenia Daskalova
9. Trust Collaboration	Yan Zheng
10. Open problems in p2p systems	Jani Lakkakorpi
11. Legal issues in p2p systems	Klaus Nieminen
12. Economics issues in p2p systems	Marcin Matuszewski

Gnutella project overview

Marina Shalamova
PhD student (student number 64301F),
Helsinki University of Technology
Riistavuorenkuja 3 B 10, 00320 Helsinki, Finland
e-mail: marina.shalamova@hut.fi,
mobile: +358 (0) 40 7493738

Abstract

In the past few years much attention is given to an information exchange framework called Peer-to-Peer (P2P). This paper focuses on the Gnutella project as a successful example of decentralized P2P network. The author gives an overview of original and present Gnutella network architecture, makes the detailed description of messages exchange within the network, provides the reader with own experience of using different Gnutella applications and depicts the main issues of the project.

1 Introduction

Exchange of files among users in the Internet has become very popular in the recent years. Many of users share and download music files, movies, software programs and other information. Software developers and researchers are trying to find new ways and to develop new techniques for reliable, efficient and secure sharing of data across the wide area networks.

In the past few years much attention is given to an information exchange framework called Peer-to-Peer (P2P). The basic premise underlying all peer-to-peer technologies is that users have something valuable to share. P2P network consists of a large number of computer nodes that are also called peers connected together. Peers may provide and consume services, they share information and services and the exchange of them is done through direct connections between peers.

Probably the earliest example of a peer-to-peer application is Zephyr chat, which resulted from MIT's Athena project in the early 1990s. Then after Zephyr chat such systems as ICQ appeared and provided a commercialized, graphical, Windows-based instant messaging system. Next was Napster, the last notable client/server-based peer-to-peer system. Gnutella and Freenet were next and led the way in decentralized peer-to-peer systems. [13]

A decentralized network has no central authority and nodes can freely join the network at any time and disconnect from the network without loss of operability of the whole system. In turn, this leads to network robustness, availability, performance, reduced cost of administration and has other benefits that attract a lot of interest from the Internet community. The decentralization of P2P network gives users the possibility to share files without storing them on central servers. Another important factor is that in P2P systems a newly joined peer brings new resources to the network. Peer-to-peer systems have shifted the Web's Client-Server model paradigm into a Client-Client model.

Beside the many advantages that have been uncovered by P2P systems, many questions and issues appear especially in the context of security and legality of sharing files.

This paper focuses on the Gnutella project as a successful example of unstructured P2P network. It is unstructured in the sense that the network consists of randomly connected hosts and shared files placed on many different hosts. The ability to have a reliable network, without dependence on any particular host, is a remarkable feature of Gnutella that has led to its immense popularity.

The paper is divided into the following sections. Section 2 gives an overview of how the Gnutella was created and developed. Section 3 depicts concept ideas of Gnutella networks, shows differences between original and present Gnutella network architectures. Next section gives a detailed description of the Gnutella protocol and depicts the whole process of file downloading, started from joining the network and finishing by closing of the Gnutella application. Section 5 shortly describes some most popular Gnutella applications. Section 6 gives an overview of Gnutella's past and present problems and of related research that will guide it on into the future. The document ends with my conclusion concerning the overview of Gnutella project and my view on its future development.

2 History of Gnutella

The first major project that implemented the concept of a decentralized file sharing system was Gnutella. Before it, systems were centralized and had one or more main control nodes. Gnutella is a real-time P2P file-sharing system that lets you search for and download files from other Gnutella users. Gnutella does not run on a server, and it is not "based" anywhere.

Gnutella has an interesting and a little bit scandalous birth history. Gnutella was born sometime in early March 2000. Gnutella originally was conceived, written, and released by Justin Frankel and Tom Pepper from the company Nullsoft, the organization that makes the Winamp player. Winamp was developed in 1999 primarily to play digital music files. According to Tom Pepper, Gnutella was developed primarily to share recipes. Two guys without college degrees developed Gnutella in just fourteen days. It was released as an experiment, but just hours after its birth the software was removed from the web site by Nullsoft's owners, America Online Inc., due to potential copyright conflicts. People were mainly using the Gnutella client for distributing copies of music files, not for sharing recipes. Thus the plans to release the specification of the protocol was given up.

Nevertheless, Bryan Mayland with some other developers reverse engineered Gnutella’s communication language and published the specification of the protocol on the Web: gnutella.nerdherd.net. There also Gnutella's Internet Relay Chat (IRC) channel, #gnutella was created. IRC channel #gnutella had a major impact on Gnutella development, particularly when rapid response from other developers was required. So, the publication of a well-defined protocol specification was extremely useful, and different developers were able to contribute their own Gnutella-compliant software that could inter-operate and soon versions of new Gnutella clients began popping up for different operating systems. Nowadays the current version of Gnutella is v0.6. [1]

Having discussed the history of Gnutella let us explain what the word Gnutella means. The name Gnutella comes from the cooperation of words GNU and Nutella. GNU is short for GNU's not Unix, the free Unix-like operating system the “geekish rallying cry of a new generation of software developers who enjoy giving free access to the source code of their products” [1, page 63]. Nutella is a delicious creamy chocolaty hazelnut spread produced by Italian confectioner Ferrero.

The question many people ask about Gnutella is, "How many users are on Gnutella?" Figure 1 shows the number of all unique hosts (green) in Gnutella network as well as the number of hosts accepting incoming connections (red).

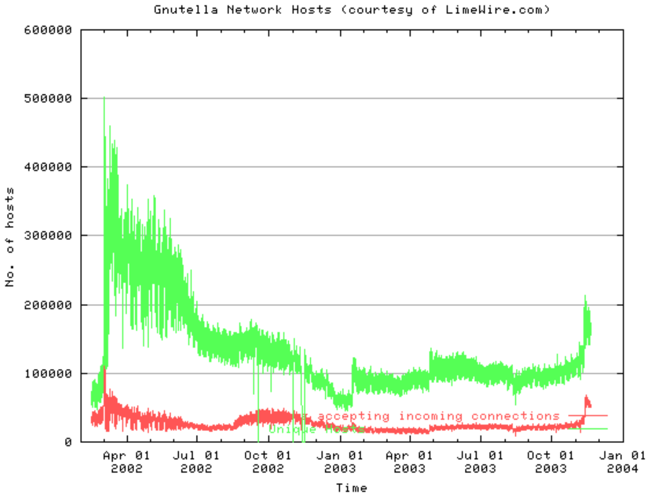


Figure 1: Gnutella Network Hosts [16]

As you can see from the figure, the number of Gnutella users is not so big nowadays, but the Gnutella community is increasing day by day. LimeWire shows that the average number of hosts found in the Gnutella network during the day of November 19, 2003 is around 180 000 hosts [16].

Due to the P2P nature of Gnutella once the new clients had been released it was impossible to stop the file-sharing network from growing in size and popularity.

3 Concept of Gnutella network

Gnutella is a decentralized peer-to-peer system. It allows the participants to share resources from their system, such as mappings to other resources, cryptographic keys, files of any type, etc., for others and also to get the resources shared by others on the network. [3]

The Gnutella protocol uses a somewhat different concept than the typical Internet client-server model. Standard network applications include three discrete modules: server, client and a network. The server contains all the intelligence. The client typically renders the result of some action on the server for viewing by the user. And finally the network connects the client and the server to allow the communication between them. Gnutella blends all these into one. "The client is the server is the network" [1, page 66]. The client and server were made as one mainly because of simplicity for users. There could be two processes, one to serve files and another to download files, but for developers it is not more difficult to combine these features in one application, and for the user one application is much easier to work with. According to LimeWare's glossary, Gnutella uses a "servent " concept where each participating peer operates as both server and client, and the name "servent" came from both of these words: SERVer and cliENT. In essence, a computer on a Gnutella network can both listen and respond when another computer talks. The interesting thing is that the network itself is embedded in each Gnutella node. Gnutella is a network built on top of the Internet entirely in software, so this is a software-based network. The Gnutella network expands with every new Gnutella node connecting to the network, and if no users run the Gnutella applications, it does not exist at all. Instead of having specialized routers, switches and hubs that enable communication, Gnutella combines all those things into the node itself.

One of the big ideas behind peer-to-peer systems is their potential to conduct information exchange without revealing their identities or even the information they are exchanging. The possibility of anonymity stems from the distribution of information across the entire network, as well as the difficulty in tracking activities on the network as a whole. Gnutella provides some degree of anonymity by enabling anonymous searching mechanism. As will be described below in the document, when searching, the user does not give to anyone the information about himself, no any IP address, e-mail or others. In most messages that are passed from node to node, there is no mention of anything that might tie a particular message to a particular user. Another benefit of Gnutella is that its routing system is not accessible from the outside. The routing tables are dynamic and stored in the memory of the countless Gnutella nodes for only a short time. Thus, it is nearly impossible to find which host originated a packet and which host is destined to receive it.

But Gnutella network is not so safe as it can be thought from the first look. For example the Wall of Shame, a Gnutella Trojan Horse, was an early attempt to nab alleged child pornography traffickers on the Gnutella network. There were some files with very suggestive filenames, which were shared by a special host. When someone attempted to download any of the files, the host would log the IP address of the downloader to a web page on the Wall of Shame. The host obtained the IP address of the downloader from its connection information. When starting to download Gnutella reveals the IP address of a downloading host to the uploading host, and vice versa. That is where Gnutella's pseudoanonymity system breaks down. [1, 6]

Another feature that some Gnutella client software implements is the notion of private Gnutella networks. To join a private network, a new node needs to know the secret handshake or password. This is a good way to ensure a high quality of service as the network has a predetermined community of members.

Section 3.1 goes in deep to original and present Gnutella network architecture.

3.1 Gnutella architecture

In Gnutella v0.4, the first public version of Gnutella, all peers are equal and connected to each other randomly. Each peer establishes and maintains connections to a few, five in average, other peers, which also recursively establishes and maintain connections to some another peers. These connections are used for sending query messages when a user is looking for some file. This message relays through the connections to all of the peer neighbours, who in turn recursively forward this query to their neighbours. The search results are then transmitted back on the same path to the originator. This scheme works fine for users with broadband connections, but not for users with slow modems because the number of query messages passed overwhelms peers with slower connections. Organizing the network in a more structured form can solve this problem. Gnutella v0.6 introduces the concept of ultrapeers. [3, 14]

The ultrapeer scheme improves network efficiency and scalability by dividing nodes into two categories: "super nodes» and "client nodes". A super node (ultrapeer) is a host with high network bandwidth connection that can act as a local hub for a large number of connecting client nodes. The super node removes the need of extensive network message routing from the client, which is a low bandwidth modem user. In such a case, the modem user uses the well-connected ultrapeer as an entry point into the network. Super nodes are connected to each other in the same way as regular peers are connected in Gnutella v0.4 network. They forward queries for their client nodes and shield client nodes from receiving unnecessary query messages. An ultrapeer only forwards a query to a client if it believes that client node can answer it. When the Gnutella network is constructed in such a hierarchal fashion, the ultrapeer concept lets the network scale quite well since it considerably reduces the number of nodes actually involved in message routing. [2, 14]

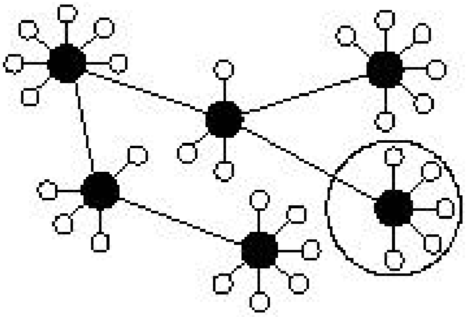


Figure 2: Gnutella v0.6 architecture [14]

Figure 2 illustrates a Gnutella v0.6 topology that was described above. Ultrapeers are depicted by dark circles, while client nodes are depicted by light circles.

Each Gnutella application after been started up, will look for other Gnutella nodes to connect to. The set of connected nodes carries the Gnutella traffic, which is essentially made of queries, reply to those queries, and also other control messages to facilitate the discovery of other nodes. Users interact with the nodes by supplying them with the list of resources they wish to share on the network, can enter searches for other's resources, will hopefully get results to those searches, and can then select those resources they want to download. Resource data exchanges between nodes are negotiated using the standard HTTP protocol. The Gnutella network is only used to locate the nodes sharing those resources. [3]

Because of Gnutella's distributed nature, an important part of any peer-to-peer application is the information exchange protocol. The Gnutella protocol is described in the next section.

4 Gnutella Protocol

Gnutella is a decentralized peer-to-peer system, consisting of hosts connected to one another over TCP/IP and running software that implements the Gnutella protocol. This connection of individual nodes forms a network of computers exchanging Gnutella traffic. Once a node has connected to the network, it communicates with its neighbouring nodes by sending and receiving Gnutella protocol messages and accepts incoming connections from new nodes that wish to join the network, for which it listens for connections on the appropriate TCP port. The Gnutella protocol defines the way the servents communicate over the network. It consists of a set of messages and also a set of rules that define the order of exchange of messages. The main Gnutella network messages are summarized in table 1. [2, 7]

Table 1: Gnutella protocol messages

Message	Description
Ping	Used to announce your presence on the network. A servent receiving a Ping message is expected to respond with one or more Pong messages.
Pong	The response to a Ping. Shows you who your active peers are. Includes the address of a connected Gnutella servent.
Query	The primary mechanism for searching files in the network. A servent receiving a Query message will respond with a QueryHit message in case a match is found against its local data set.
QueryHit	The response to a Query. This message provides the recipient with information needed to get the data matching the corresponding Query.
Push	A mechanism that allows a servent that is under a firewall to contribute files to the network.
Bye	An optional message used to inform the remote host that you are closing the connection, and specify your reason for doing so.

Gnutella is a broadcast-type network, in which Pings and Queries are duplicated and forwarded to multiple other nodes. Pongs and QueryHits messages are routed by each node back along the path the corresponding request came to the node. In this respect, requests are very inefficient due to the flood nature, but replies are directed back to unique nodes rather efficiently. [6, 12]

To avoid continuous repeating of a message, a message carries a 128-bit unique identifier named Unique Universal Identifier (or UUID). Every time a message is delivered or originated, the UUID of the message is memorized by the host that is passing the message. If there are loops in the network then it is quite possible that a host could receive the same message twice. Therefore, if a message with the same stored UUID is received again at a later time, it is not retransmitted. This explicitly prevents wasting network resources by sending a query to hosts that have already seen it.

Another interesting idea that Gnutella implements is the idea of using time-to-live number (TTL) to control routing. Each message has a TTL number. Typically a query starts life with a TTL of 7, which is decremented when a message passes from host to host. When the TTL reaches 0, the request has lived long enough and is not retransmitted again. This technique also helps to prevent the flood of Gnutella protocol messages. [1]

4.1 Connecting to the Gnutella network

A node that wishes to participate in the network has to join the Gnutella network by finding an initial host to start its first connection. Currently, mechanisms known as "host caches" allow new nodes to easily participate in the network. There are several permanent Gnutella hosts whose purpose is to provide a list of Gnutella hosts to any Gnutella servent connecting to them. A list of known public host caches is available at: http://groups.yahoo.com/group/thegdf/database?method=reportRows&tbl=5 (requires Gnutella Development Forum membership). Since host caches can be overloaded, or go down, a servent must not be dependent on them. A popular technique is to keep a list of a few hundred hosts on the hard disc of your computer. The probability that at least one of them is running is very high, even if the list is several weeks old.

A new node opens a TCP/IP connection to an existing node and performs a handshake that includes an exchange of servents capabilities, since different Gnutella clients may choose to implement different features. If the server decides to refuse a new client (e.g., with the 503 Busy response), it is encouraged to provide an X-Try: header that recommends other IP/port addresses to try connecting to. Once the first connection is established, the addresses of more hosts will be supplied using Pong messages. [3, 17]

Figure 3 illustrates the process of joining a new node A to the Gnutella network.

A User A performs the following steps in order to join the Gnutella network and become its active participant:

1. User A uses host caches technique discussed in this section to join the Gnutella network. In this case, the user connects to a GnuCache server. There is also possibility of using Web or IRC for the same purpose.
2. GnuCache returns a list of nodes on the Gnutella network, from which User A chooses one (User B) and attempts to contact it.
3. User A sends a "Gnutella Connect" to User B to request to join the Gnutella network: GNUTELLA CONNECT/<protocol version string>\n\n, where <protocol version string> is the ASCII string “0.6” i.e. the current version of the specification.

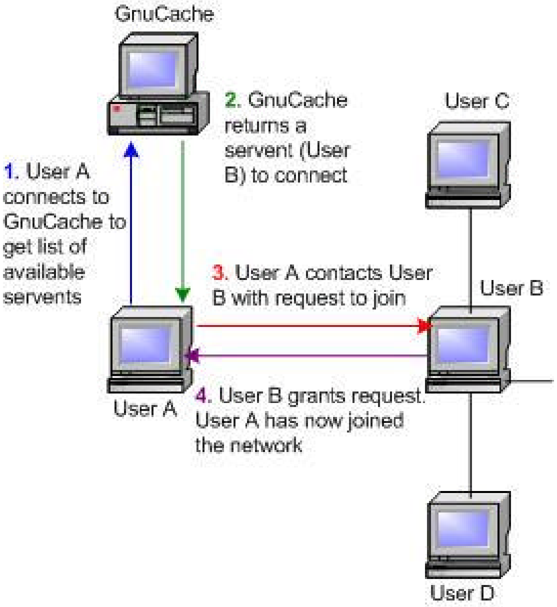

GnuCache
2. GnuCache returns a servent (User B) to connect
User C
1. User A connects to GnuCache to get list of available servents
3. User A contacts User B with request to join
User B
User A
4. User B grants request. User A has now joined the network
User D

Figure 3: New node joins the Gnutella network

4. User B accepts and returns a "Gnutella OK" to User A. User A is now part of the Gnutella network.

Once a servent has connected successfully to the network, it communicates with other servents by sending and receiving Gnutella protocol messages.

A Ping message is used to announce your presence on the network. When another computer receives your Ping it will respond with a Pong message. It will also forward your Ping packet to other computers to which it is connected and, in response, they also will send back Pong messages. Each Ping and Pong message contains a Globally Unique Identifier (GUID). A Pong also contains an IP address, port number, and information about how much data is being shared by the computer that sent the Pong message. Pong messages are not necessarily returned directly to the point of origin, instead they are sent from computer to computer via the same route as the initial Ping. Thus, after sending a Ping to one computer you will start receiving many Pong responses via that one computer. When sending a Ping message, servent cannot know if it will reach only the neighbour host, or many hosts on the network. It depends on what system for handling Ping and Pong messages other servents are using.

To reduce the traffic of Pong messages the pong-caching scheme is used. There are several schemes of handling Ping/Pong messages. Most of them are based on the TTL value and Hops value – the number of hosts the packet has passed through. The basic idea of one such scheme is that the servent (say A) remembers all Pongs it has received through its current connections. When another servent (say B) sends a Ping message to A with TTL equal to N, A still replies with the usual Pong, but also sends all its cached Pongs with hops count less than N without broadcasting the Ping. Every few minutes or so, each servent also sends a new Ping to its neighbours to update its pong cache. Other Ping/Pong caching schemes are available at Gnutella Development Forum http://groups.yahoo.com/group/the_gdf/files/Proposals/PONG/Variants/pingreduce.txt.

So, now you know who your active peers are, and you can start making searches. [6]

4.2 Searching in the Gnutella network

Users of the Gnutella network can perform searching of some content by sending arbitrary queries into the system. Gnutella "Query" messages allow you to search by asking other computers if they are sharing specific content. The node that wishes to begin the search sends a Query message to all nodes connected directly to it, so to its neighbours. Each of these nodes in turn replicates and sends the Query message to its own neighbours, except the node the query came from. Servents that use Ultrapeers technique will not always forward every Query over every connection.

Because queries occupy so much of the Gnutella network bandwidth, the specification restricts the maximum size of a Query message to 256 bytes. Additional measures are checked that the queries do not uselessly and endlessly traverse the network. Gnutella messages have associated TTL values, which are decremented at each hop. Nodes silently remove the query once the TTL field reaches 0. Nodes are also responsible for dropping queries that they see twice. Queries include a "minimum speed" field, which directs nodes to reply to the query only if they can satisfy the minimum transfer speed for file transfers. The Query also consists of a "search criteria" field and extensions block, allowing clients to search for files based on a number of criteria, sometimes client-specific. The query is based on a series of keywords, meant to locate files that match all keywords. Additionally, a special type of query exists which requests a list of all files served by a node. [2, 6]

A node replies with a QueryHit message when it has content that satisfies the searching criteria in the request. Most importantly, the QueryHit contains an IP address and port number where this specific node can be reached via TCP connection for the actual file transfer. These Query Hit messages are only sent along the same path that carried the incoming Query message. [2, 3]

A servent may also create Queries automatically, to find more locations of a resource for example. If doing so, the servent must be very careful not to overload the network. A servent should not send more than one automatic query per hour. [3]

4.3 Gnutella file transfer

Once a node receives a Query Hit message that satisfies its request, it knows where to find the file it wants. An important point about Gnutella is that files are downloaded out-of-network: the two hosts involved in the transfer connect over TCP/IP and transfer the data directly, instead of wasting the Gnutella network capacity with bytes from files.

File data is never transferred over the Gnutella network. The file is downloaded using the HTTP/1.0 or the HTTP/1.1 protocol, the standard protocol for downloading files from web servers. The details of HTTP are out of scope of this paper, and can be found in RFC 1945 and RFC 2616 for versions 1.0 and 1.1 respectively. The node wanting to download a file makes a TCP/IP connection to the serving host at the IP address and port number specified, then makes a standard HTTP request. This scheme is modified slightly if the serving host is firewalled as will be described in the next section. [2, 3]

In the new versions of Gnutella clients, an "active queuing mechanism" is implemented allowing upload transfers to be queued and undertaken at a later time. This mechanism provides the queued recipient with a live update of their positions while moving toward the head of the queue. The mechanism described here is very simple. Although a number of alternative proposals have been made to provide upload queuing support.

Some key requirements met by this mechanism are:
- Upload queues should provide both users with a visual indication of the queue position.
- They should not rely on a "reverse connect", which may not be possible because of a firewall.
- Holding a position in the queue should require maintaining an "idle" connection: if you drop the connection, you will lose your place.

Upload queues are supported through a single additional header "X-Queue", which is included in both the HTTP request and response. Clients which support queues send "X-Queue: 0.1", which simply tags the request as a candidate for queuing. If this header is not received, the requesting client is assumed to follow normal Gnutella behaviour in the case of a busy response. If there is an upload slot available, the download begins as normal. If not, the request is placed at the end of the queue and a 503 response is returned with the additional X-Queue header, which by form is as follows [8]:

X-Queue:
position=2,length=5,limit=4,pollMin=45,pollMax=120

Every part of this header is optional, and if desired it can be broken into multiple headers, etc. The "position" key indicates the request's position in the queue, where position 1 is next in line for an available slot. The "length" key indicates the current length of the queue, and used just for informational purposes. Likewise the "limit" key specifies the number of concurrent uploads allowed. All of this information is completely optional and only used for displaying to the client. Finally, "pollMin" and "pollMax" provide hints to the requesting client as to how often it should re-request the file (in seconds). Requesting more often than pollMin will be seen as flooding, and cause a disconnection. Failing to issue a request before pollMax will be seen as a dropped connection. If these items are not present in the header a default retry interval can be used.

This approach has some key advantages:
- The downloader can see the change of their place in the queue as they move towards the first position, so even if the queue is long, at least progress can be observed. It is important not to underestimate the value of showing visible progress to the user.
- Because the HTTP request is reissued periodically, the client is able to request the most appropriate "Range" each time.
- By requiring from the requesting client to maintain a connection, there is no need to hold open upload positions for a request that may never come. If the client is no longer interested in downloading from this source, it can close the connection immediately.

Upload queues represent an important step for the evolution of Gnutella because they reward users who have waited for a file, rather than an earlier approach which rewarded users who abuse the system by requesting too often. It is also much more satisfying for a user to see a decrementing queue position which assures that progress is being made, rather than a seemingly never-ending stream of busy messages. [8]

4.4 Using of PUSH message

It is not always possible to establish a direct connection to a Gnutella servent in order to initiate a file downloading. The servent may, for example, be behind a firewall. Firewall-friendliness is a difficult problem to solve, because Internet users who are behind firewalls cannot accept incoming TCP connections from the outside world[1]. They can certainly join P2P networks by connecting to other hosts, and they can exchange network messages over the established TCP connections, but they cannot themselves allow other Internet hosts to connect to them in order to join the network. This fact challenges the growth of P2P networks in the long term.

File transfers can be a problem if two hosts have to contact each other directly. If the host serving the file is behind a firewall, other nodes cannot connect to it for downloading the file. Normally, your computer will initiate the HTTP connection to the computer that has the file. A partial solution offered by Gnutella is the Push message. The node that wants to download data from a firewalled host sends a Push message through the Gnutella network to that host. Push allows a message to be delivered to the computer that has the file you would like to download via the route that the QueryHit packet originally travelled, except in reverse. The Push message tells that computer that you would like to download a file but cannot manage to initiate an HTTP connection. The computer then becomes the initiator of a connection directly to you, which often is possible because the firewall between the machines is only limiting connections initiated from outside the firewall. If the node that wants the file is itself not behind a firewall, the file transfer can then proceed. [2, 3, 6]

If the direct connection cannot be established anyway, then it is likely that the servent that issued the Push request is itself behind a firewall. In this case, file transfer cannot take place by the means of what is described in this document. If a direct connection can be established from the firewalled servent to the servent that initiated the Push request, the firewalled servent should immediately send the following:

GIV <File Index>:<Servent Identifier>/<File Name><lf><lf>

Where <File Index> and <Servent Identifier> are the values of the File Index and Servent Identifier fields from the Push request received, and <File Name> is the name of the file in the local file table whose file index number is <File Index>. The File Name may be url/uri encoded. The servent that receives the GIV (the servent that wants to receive a file) should ignore the File Index and File Name, and request the file it wants to download. The servent that sent the GIV must allow the client to request any file, and not just the one specified in the Push message.

[1] Editors note: being behind a NAT or worse two NATs has the same effect unless a special NAT traversal technique is used.

If the TCP connection is lost during a Push initiated file transfer, it is strongly recommended that the servent who initiated the TCP connection (the servent providing the file) attempt to re-connect. That is important, since the servent receiving the file might not be able to get another Push message to the servent providing the file. [3]

4.5 Last sent packet

The Bye message is an OPTIONAL message used to inform the servent you are connected to that you are closing the connection.

It is safe to retrofit the Bye packet into the v0.4 protocol. Indeed, the packet is the last message that will be sent by a disconnecting servent, and otherwise it will be ignored as a bad packet by older servents, which is not dramatic. This message is thought to be useful to the developers of Gnutella servents. Users will be able to report errors, and maybe understand what is going wrong, with them or with the remote node. Bye message is also considered as carrying some "social value". Friends say good-bye to each other when they part, and the Gnutella network is some kind of modern electronic friendship, where people gather and friendly share and exchange files.

Servents should send a Bye message to a node as the last thing on the network, and then close the connection. A Bye packet must be sent with TTL=1 to avoid accidental propagation by an unaware servent. The data may not be delivered to all neighbours, but at least the servent tried. Upon reading a Bye packet, a node should immediately close the connection and stop processing any other received messages from that connection that were still pending processing. [10, 3]

The servent that sent the packet must wait a few seconds for the remote host to close the connection. Any other data must not be sent after the Bye message. After sending the Bye message, and during the "grace period" when we do not immediately close the connection, the servent have to read all incoming messages, and drop them unless they are Query Hits or Push, which may still be forwarded. The connection will be closed as soon as the servent gets an EOF (End Of File) condition when reading, or when the "grace period" expires. [3]

5 Gnutella applications

This section describes various implementations of peer-to-peer clients based on the Gnutella open protocol. Table 2 summarises all the most popular existing Gnutella applications.

Table 2: Gnutella clients for Windows, Linux/Unix, and Macintosh [18]

Windows	Linux/Unix	Macintosh
BearShare Gnucleus LimeWire Phex Swapper XoloX	Gtk-Gnutella LimeWire Mutella Phex Qtella	LimeWire Phex

Since all applications are based on the same protocol, they provide a similar core, however, different features are included in each program.

The differentiation of the applications is based on the following parameters:

1. Graphical User Interface (GUI)
2. Additional features
3. Easiness of installation

Also, these clients are differentiated by the platform for which they are implemented.

5.1 Windows based Gnutella clients

This section gives quick overview of Windows based Gnutella compliant clients.

BearShare

The BearShare client was developed by FreePeers Inc. At the moment, there are two versions available free as well as the commercial BearShare Pro version. The overall experience from using BearShare is the following:

- Intuitive Windows XP like GUI has not brought any problems, altogether with good customer support and user forums this client deserves high points for the user interface (UI).
- As additional features the BearShare supports advanced search, automatic resume of file downloading, chat and forum access, swarming (ability to download a single file from multiple hosts in parallel), parental control to prevent porno downloading by children, media player, file verifier, bandwidth and update options.
- During the installation no problems have been found.

LimeWire

This client supports Windows, Macintosh as well as Linux platforms. As in the BearShare case, both free and priced clients are available. Using similar parameters the grade is the following:

- The UI's look and feel can be easily changed, as skinning support is provided. The main controls are logically grouped on the main screen thus first impression is that the UI is overloaded with information; however after couple of hours I found it quite useful and faster to navigate than in BearShare.

- The following features of the LimeWire can be mentioned: ultrapeer technology, media player, chat, skins, swarming, local network searches and magnet links support. Magnet links allow website designers to provide links to files that can be downloaded with peer-to-peer technology. The use of magnet links can speed up popular downloads for end users.
- The installation process is as simple as in the previous case.

Other Windows clients

As other clients for the Windows operating system the following can be mentioned: Gnucleus, Phex, Swapper, XoloX. They are Gnutella protocol compliant, but lacking features compared to BearShare and LimeWire applications.

5.2 Linux based applications

This section represents main applications for the Linux/Unix platforms.

GTK-Gnutella

The GTK-Gnutella has a very complicated GUI that consists of multiple tab views together with dozens of unneeded controls.

While the UI is not so attractive, the GTK-Gnutella provides different useful features like: swarming, passive search, searching by URN (Universal Resource Name), SHA1 (Secure Hash Algorithm v.1) and magnet as well as it is compliant to:

- HTTP/1.1
- Gnutella v0.6
- GNet compression
- GWebCache Proposal
- HUGE (Hash/URN Gnutella Extensions)
- PARQ (Passive/Active Remote Queuing)
- PFSP (Partial File Sharing Protocol)
- QRP (Query Routing Proposal)

As pre-requisites, GTK must be installed on the machine before installation of the GTK-Gnutella client.

QTella

The QTella is a QT based client that is distributed free of charge.

- UI provides nice looking controls as well as a possibility to change the look and feel. In general the UI is complicated and consist of tens of tab views.
- The QTella provides a basic set of features for the user, but the Sharp Zaurus version can be noticed as an advanced modification of this client.
- In order to run this client the user should have preinstalled the QT frameworks.

6 Gnutella issues and optimization techniques

Several problems can be defined at the Gnutella network, but I think that the most important are: spamming, free riding, reliability issues and security problems.

6.1 Spamming

Spamming is well known in the E-mail World. We all from time to time receive unsolicited e-mails, with advertisements or sometimes, even viruses. It is obvious that spamming is probably one of the most difficult things to solve in P2P file-sharing systems such as Gnutella. All messages in the Gnutella network are sent in plain text, readable and modifiable by everyone. Serious cases of spam can happen because some nodes may just return a commercial message on each and every search query.

If we accept to download a spammed query result we might become an active part of a Distributed Denial of Service (DDoS) attack against some Internet Host. Gnutella has no provisions to thwart Distributed Denial of Service Attacks. DDoS attacks are probably the first thing that comes to ones mind when we think of a huge number of "uncontrolled" participating hosts. Unfortunately, no provision has been taken to avoid such attacks, neither in the protocol nor in the majority of Gnutella clients. [12]

6.2 Free riding

A lot of attention has been focused on the copyright laws and free access to any kind of music as P2P networks are usually used for exchanging musical files. In such large systems, where the users are not monitored as to who makes their files available to the rest of the network or downloads remote files, another problem is also very actual, namely - free riding. As the user community in such networks gets large, users will stop producing and only consume. This free riding behaviour is the result of a social dilemma that all users of such systems confront, even though they may not be aware of its existence.

Since files on Gnutella are treated like a public good and the users are not charged in proportion to their use, it appears rational for people to download music files without contributing by making their own files accessible to other users. Because every person can reason this way and free ride on the efforts of others, the whole system's performance can degrade considerably. [9]

6.3 Reliability issues

Due to the constant changing of the Gnutella network structure (peers are continuously joining and leaving the network), it is very difficult to ensure a connection between two given machines. It is even possible for a node to completely drop out of the network, although that is unlikely. What is more, due to the limited network bandwidth and processor power, search queries get dropped frequently. This typically happens when a Gnutella message has to pass trough a node, which cannot handle the huge amount of traffic. Thus, the Gnutella protocol doesn't guarantee any reliability in any form.

6.4 Security problems

Security is not a part of Gnutella protocol. It would be very hard to embed conventional security technologies into the protocol as every node through which the message passes, should be able to read the full content of the message. Thus, for example, encryption does not have any sense.

Another huge security problem in the P2P community is spyware programs. A spyware is a program that is usually distributed along with the P2P client and which sends out personal user information. Cydoor is an example of such a program, which is distributed along with some of the most popular P2P clients.

Viruses are another potential thread for the P2P community, and P2P systems are just another medium over which viruses can spread efficiently. It is estimated that they are not as dangerous as viruses, which are spread over e-mails, but still it is a big problem. Most Anti-virus vendors have already released protection updates for the well-known Trojan horses VBS.Gnutella and W32.Gnuman.Worm.

Finally, poorly written P2P Clients are another problem in these networks. For example, the actual transfer of files in Gnutella is done with the HTTP protocol. This means that each Gnutella client instantiates also a mini web-server. These mini web-servers are sometimes poorly written making them vulnerable to attacks. [12]

The Gnutella Development Forum suggests some complementary protocols to Gnutella in order to build a secure peer-to-peer network. The proposal is divided in three main protocols: Gnutella Certification Acquisition Protocol (GCAP), Gnutella Conflict Resolution Protocol (GCRP) and Gnutella Circle Checking Protocol (GCCP).

6.5 Gnutella Certification Acquisition Protocol (GCAP)

This protocol establishes a mechanism of identifying individual users by certificates. Those certificates use e-mail addresses for acquisition but the addresses could be anonymous (they use a GUID for identification). The goal of this protocol is to establish a certain node identification scheme within a Gnutella network, and not to identify people. The e-mail mechanism is useful because an attacker or hostile don't have an infinite number of e-mail to retry, but it is not so costly to any user, who even could create a mostly anonymous account in any public web e-mail service. Giving e-mail for registration is also a common Internet practice. [11]

6.6 Gnutella Conflict Resolution Protocol (GCRP)

The GCRP protocol used for resolution of conflicts between servents when one servent thinks that another has a bad behaviour. It uses an opinion that a mass number of different servents complaining about some other node can't be wrong, because they are precisely the network. This means that an attacker or hostile node who have an impact in a lot of nodes will be subject of a lot of complaints. So an attack, which affects a few nodes, will be ignored, as that attack is less important to the network (if an attack or abuse affects a few nodes at a time), but if the attacker keeps doing it, anyway it will generate a lot of complaints. So, in order to be effective, a complaint against an attacker must come from several hundreds of hosts [11]

6.7 Gnutella Circle Checking Protocol (GCCP)

This protocol is only applicable to ultrapeers; client nodes will work the same way as present. This protocol assumes that an ultrapeer has a very limited number of connections to other ultrapeers, usually three. The other servents will reject any *s* hostile event, which have or try to have a large number of ultrapeer connections, and a complaint (see Section 6.4.2) is generated about it.

The goal of this protocol is to get a node surrounded by a circle of nodes, which check any message with the other's signature. So, it is not possible for a node to modify a relayed message. A way of breaking this circle check will be two malicious nodes working together. But the protocol makes the hostile's task more difficult. [11]

The more detailed information about these protocols can be found at GDF, http://groups.yahoo.com/group/the_gdf/files/Proposals.

7 Conclusion

In this paper, Gnutella project has been described. Gnutella is a search protocol that allows peers to search without the need for any centralized control. In Gnutella every node in the network is both a client and a server and is called servent. Servents join the network using one of several techniques e.g. the Web, IRC, host cache method and once having joined it can discover other peers through the use of the Ping/Pong messages. The Gnutella protocol defines the way servents communicate over the network. It defines a set of Gnutella messages and a set of rules governing their inter-servent exchange. The present Gnutella network architecture differs from the original one. Currently an ultrapeer scheme is used to make more efficient usage of the network bandwidth. In spite of all the P2P network benefits, the Gnutella also has some weak factors, for example spamming, free riding, reliability issues and security problems. Gnutella protocol is an open one that makes it possible for software developers to create their own Gnutella applications. In this document the most popular Gnutella clients were described.

References

[1] Oram, Andy (editor): Peer to Peer: Harnessing the Power of Disruptive Technologies, First Edition March 2001, ISBN: 0-596-00110-X, Chapters 8 and 14

[2] Berkes, Jem E.: Decentralized Peer-to-Peer Network Architecture: Gnutella and Freenet, University of Manitoba, Canada, April 9, 2003, http://home.cc.umanitoba.ca/~umberkes/filesharing.pdf (accessed November 12, 2003, requires GDF membership)

[3] Klingberg, Tor and Manfredi, Raphael: Gnutella 0.6 (draft), Network Working Group, June 2002, The Gnutella Developer forum (GDF), http://groups.yahoo.com/group/the_gdf/files/Development (accessed November 13, 2003, requires GDF membership)

[4] Bildson, Greg and Rohrs, Christopher: An Extensible Handshaking Protocol for the Gnutella Network, Lime Wire LLC, September 2001, Gnutella Developer Forum (GDF) http://groups.yahoo.com/group/the_gdf/files/Proposals (accessed November 13, 2003, requires GDF membership)

[5] Clip2: The Annotated Gnutella Protocol Specification v0.4, Document Revision 1.8, The Gnutella Developer Forum (GDF), July, 17 2003, http://groups.yahoo.com/group/the_gdf/files/Development (accessed November 13, 2003, requires GDF membership)

[6] Batkins, Brenda L.: An Overview of Gnutella, Version 1.2e, July 27, 2001, http://www.sans.org/rr/papers/60/455.pdf (accessed November 17, 2003)

[7] Coman, Alex: Analyzing a Gnutella Client (Project report), University of Alberta, 2003, http://www.cs.ualberta.ca/~kenw/courses/2003/cmput664/project/student/acoman.pdf (accessed November 15, 2003)

[8] Stokes, Michael: Active Queuing Mechanism, The Gnutella Developer forum (GDF), August 10, 2002, http://groups.yahoo.com/group/the_gdf/files/Development (accessed November 13, 2003)

[9] Huberman, Bernardo A. and Adar, Eytan: Free Riding on Gnutella, First Monday, volume 5, number 10, October 2000, http://firstmonday.org/issues/issue5_10/adar/index.html (accessed November 20, 2003)

[10] Manfredi, Raphael: Introducing New Bye (0x02) Packet in Gnutella 0.4, The Gnutella Developer Forum (GDF), March 15, 2002, http://groups.yahoo.com/group/the_gdf/files/Development (accessed November 13, 2003)

[11] Maeso, Gregorio: General Security Proposal, The Gnutella Developer Forum (GDF), 2002, http://groups.yahoo.com/group/the_gdf/files/Development (accessed November 13, 2003)

[12] Zeinalipour-Yazti, Demetrios: Exploiting the Security Weaknesses of the Gnutella Protocol, University of California, March 2002, http://www.cs.ucr.edu/~csyiazti/courses/cs260-2/project/gnutella.pdf (accessed November 19, 2003)

[13] Blundell, Nick and Mathy, Laurent: An Overview of Gnutella Optimisation Techniques, Lancaster University, 2002, http://www.comp.lancs.ac.uk/computing/users/blundell/DeptSite/public/PGNet2002Presentation.ppt (accessed November 18, 2003)

[14] Atip Asvanund, Sarvesh Bagla, Munjal H. Kapadia, Ramayya Krishnan, Michael D. Smith, Rahul Telang: Intelligent Club Management in Peer-to-Peer

Networks, May 2003, http://www.sims.berkeley.edu/research/conferences/p2pecon/papers/s6-asvanund.pdf (accessed November 12, 2003)

[15] Singla, Anurag and Rohrs, Christopher: Ultrapeers: Another Step Towards Gnutella Scalability, November 26, 2002, http://www.limewire.com/developer/Ultrapeers.html (accessed November 20, 2003)

[16] LimeWire: Running on the Gnutella Network, http://www.limewire.com/english/content/netsize.shtml (accessed November 20, 2003)

[17] Taylor, Ian: Lecture 3: Gnutella, Distributed Systems, 2003, http://www.cs.cf.ac.uk/user/I.J.Taylor/DistributedSystems/lecture3.pdf (accessed November 20, 2003)

[18] Gnutelliums - Gnutella Download, http://www.gnutelliums.com (accessed November 20, 2003)

The Freenet Project: Signing on to Freedom

Renjish Kaleelazhicathu,
Networking Lab, HUT
renjish@netlab.hut.fi

Abstract

Freedom of expression has been widely accepted as a fundamental right of every human being. Internet currently provides this freedom to a greater extent than any other medium. However, there is erosion in privacy and censorship in recent years and this is expected to get worse in the future. Recently, efforts have been made by the Internet community to protect privacy mainly by using peer-to-peer technologies. The paper summarises one such project, namely, the Freenet project, which aims to create an uncensored and secure global information storage system over the Internet. The Freenet project's architecture, security, usability aspects and performance related issues are also discussed in detail. Conclusions are drawn based on its comparison with other publishing systems like Free Haven. The paper also suggests the required improvements.

Keywords: *Peer-to-peer, Freenet Project, GUID, Keys.*

1 Introduction

Recent times have witnessed an increasing threat to privacy in the cyber space. While censorships are necessary in maintaining law and order in a society, a misuse of such rules has become an increasing cause of concern among the cyber citizens. Peer-to-peer (P2P) technology [1], an old yet recently much popular technology has come to the rescue in this regard. Progress in the processing power, storage area and other aspects of personal computers have made the implementation of P2P much more feasible in the present.

The Freenet Project [2] aims to achieve freedom of expression online using P2P, in order to create and maintain a global virtual file system. It has a completely decentralized architecture and supports scalability and fault tolerance. The architecture maintains data integrity and prevents privacy violations.

The organisation of the paper is as follows. Section 2 describes the related work in this area. Design challenges are discussed in section 3. Section 4 introduces the concept of Freenet project. Its architecture that includes the data operations and the routing scheme are dealt in section 5. Section 6 looks at security issues. Performance analysis is discussed in section 7. Section 8 describes the ongoing improvements done to Freenet while section 9 briefly describes its usability aspects. This is followed by conclusions in section 10.

2 Related work

Much work has been done in the past that provides some features offered by the Freenet architecture. While some of these schemes are complementary to Freenet, some fail to provide complete anonymity that is crucial for maintaining privacy online and is a fundamental design goal of this architecture. The following are some of those schemes and proposals.

Chaum's mix-net scheme [3] helps to create anonymous point-to-point channels. This has been used for emails by Mixmaster remailer [4] and in case of TCP/IP by onion routing [5]. However, this scheme doesn't enable one-to-many publication and also doesn't support file access and storage.

In web services, browser proxy services like Anonymizer [6] tend to provide anonymity to the customers while giving little protection to the producers of the content. It also doesn't provide protection from the services maintaining logs of these customers.

Publius [7] is yet another publishing system that enhances availability by maintaining redundant shares of files among *n* servers, *k* of which are only required to reconstruct it. However, these servers themselves being well known are vulnerable to attacks. Free Haven [8] is similar to the Freenet initiative and supports anonymity, accountability, persistence and flexibility. Some of the other related approaches are distributed.net [9] for sharing CPU cycles, Napster [10] and Gnutella [11] for file sharing and Akamai [12] for file replication for the corporations. None of these approaches provide anonymity. Thus, there definitely arose a need to have a publishing system that could address the following design challenges.

3 Design Challenges

The challenges to be taken care of in designing a publishing system were to provide anonymity for the producers and consumers, online security against malicious attacks, scalability, fault-tolerance and higher availability.

4 The Freenet Project

The Freenet Project was born as a response to the threat against freedom of expression online. Ian Clarke, then a student at University of Edinburgh initiated this project with his paper [13] in 1999. Since then he has managed the project with contributions from the Internet community. More information on the source code and installation of Freenet is available at [14]. Freenet is a publishing system that provides all the participants an opportunity to publish and read information on the Internet with complete anonymity.

4.1 Design Goals

The underlying idea of this publishing system is to provide freedom of speech and expression on the Internet. Anonymity plays a crucial role in achieving this goal. All the design challenges mentioned in section 3 were taken care of while developing the architecture of Freenet. This is an ongoing project and new additions and features can be expected by keeping the underlying design goals intact. The following summarises those goals [15]:

- Privacy for information producers, consumers and holders.
- Resistance to information censorship.
- High availability and reliability through decentralization.
- Efficient, scalable, and adaptive storage and routing.

5 Architecture

As mentioned before, Freenet architecture is based on the P2P concept. The participants of Freenet mainly share storage space unlike grids, which share CPU cycles and some other P2P applications that share files. They conduct all their operations on the files viz. search, storage, management and retrieval using location-independent globally unique identifier (GUID) keys.

Hence, key generation plays a central role in the Freenet architecture. A GUID is a binary key obtained by applying a hash function. The 160 bit SHA-1 [16] is used as the hash. There are two main types of keys used. *Content-hash keys* (CHKs) for primary data storage and the *signed-subspace keys* (SSK), which are for higher-level human use. An explanation of these keys is as follows:

- *Content-hash key (CHK)*

This is the low level data storage key generated by hashing the file contents. This method of key generation helps to fuse multiple files with same content, since the keys generated from files with the same content will always be the same. CHK provides a unique identification to every file. CHKs are useful for updating a file as well as splitting a large file into multiple parts. These are explained in detail in later sections.

- *Signed-subspace key (SSK)*

This key sets up a sub-space readable by any user but can be written only by the owner.

The subspace for an archive can be created as follows. The first step in creating an archive for a paper on Freenet is to generate a random public-private key pair that would identify it. In order to insert a file, a text description of the file is chosen, for instance, `paper/Freenet/architecture.` The SSK for the file can be generated by first hashing the public half of the subspace key and the descriptive string independently and the output of these are then added and hashed again. The file is signed with the private half of the key in order to provide an integrity check so that any node that handles a signed sub-space file validates its signature before accepting it.
An illustration of the SSK generation is shown in figure 1.

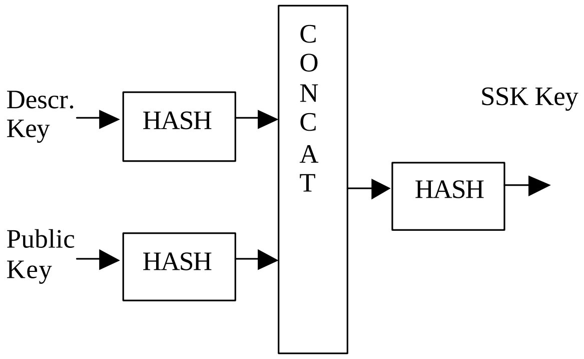


Figure 1 SSK Generation

The SSKs and CHKs are analogous to the filenames and inodes in Unix file systems. The SSK provides an indirect reference to the file by pointing to the CHK which uniquely identifies each file.

5.1 Routing

An important requirement of Freenet is to route queries to the data or the file efficiently. Napster uses the centralized P2P architecture to accomplish this and Gnutella follows the flooding scheme. Both schemes have issues. The centralized scheme in Napster provides a single point of failure while the flooding in Gnutella causes wastage of bandwidth and is thus unscalable.

Freenet has adopted the steepest-ascent hill-climbing search to avoid both of these problems. In this scheme, each node forwards the query to the node which it feels is closer to the destination.

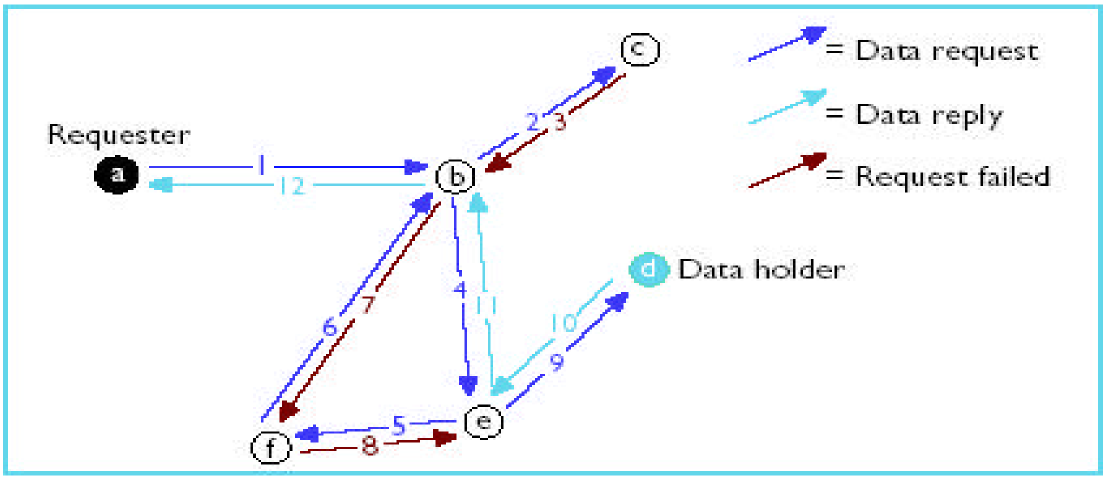


Figure 2 A Request Sequence

5.1.1 File Retrieval

Each routing table entry of a node in Freenet maintains the GUID keys of other nodes and their addresses. When a node receives a request, it first checks its own storage and attaches the tag identifying itself as the data holder in case the data is found. Else, the query is forwarded to the next node in its table that has a key closer to the query. The process of forwarding at each node continues as before until the query reaches the destination. Upon reaching the destination, the data is sent back the chain to the requester and subsequently creates a new entry in routing table with the data holder's key and address. Each node could also store this data in its local cache before forwarding it back to the recipient.

To maintain privacy, each node could place its own tag as data holder on the reply message as it is relayed back to the requestor while subsequently storing the real data holder's details in its routing table. This would help to keep the data holder anonymous. The recipient avoids the case of looping by bouncing it back to the sender. If a node fails to find any node to forward further, it notifies the failure back to its previous node.

An instance of a file retrieval sequence is illustrated in figure 2. In order to limit resource usage, the time-to-live (TTL) field is decremented at each node before forwarding.

This file retrieval approach brings the destination closer with each hop. A subsequent query would tend to approach the previous request's path and can use the locally cached copy once the paths are converged. Nodes that answer the queries reliably are added to more routing tables and hence are contacted more, thus adding reputation to that node.

5.1.2 File Insert

The file insert message follows the same scheme as is taken by a file retrieval message. A user assigns a GUID for the file to be inserted and sends it to its own node with a TTL value that represents the number of copies of the file to be stored on the network. An insert fails if CHK or SSK is already present. In case of the existence of SSK, a new description string has to be chosen again or the user should perform an update rather than an insert.

If the key doesn't already exist, it is forwarded to the nodes with GUID key values closer to that in the message until the TTL becomes zero and an "all clear" message is sent back to the user. The user then sends the file through the path established and each node stores the file after verifying it against its own GUID and creates a routing table entry with the data holder as the final node in this chain. Any looping or failure to find the next node is dealt in the same way as in file retrieval.

5.2 Operations

5.2.1 The Search

A major design issue is to find a way for the user to search the network for a relevant file key. The simplest way to add search capability to Freenet is by running a hypertext spider as used on a web. However, this provides a centralized solution and hence is not in accordance with the design goals. Yet another method is to create a series of lightweight indirect files carrying pointers to the real file and have names according to the search keywords chosen by the user. For instance, while inserting a paper " The Freenet Project", the user can add several indirect files with names like `keyword: technology` and `keyword: Internet.` Multiple such indirect files with the same search keyword can exist unlike in the case of a real file. Also, on a request based on a certain keyword, say technology, the results would show multiple real files along with "The Freenet Project" paper. However, managing the large volume of such indirect files is yet another issue.

5.2.2 Storage Management

The free nature of Freenet demands better storage management in order to maintain greater availability of data in the network. This is currently achieved by prioritising the space allocations by popularity of the files. The popularity is defined by the rate of requests for a file. The least recently requested files are deleted at a node if the demand for space arises. Since routing table entries are smaller, the files can still be received from the data holder if required. The original data holder has always a greater probability of having the file. This is because Freenet's data holder pointers have a tree-like structure. While the nodes at the leaves will see few requests, those higher up in the structure receive more requests from the network thus maintaining the copies for a longer period of time.

Hence, the file distribution is dependent on two components: tree growth and pruning. The query-routing mechanism enables more copies of a file to be created in an area of the network where the request arises thus resulting in tree growth. Copies of the files that are least requested in other parts of the network are deleted resulting in pruning. Thus the number and location of the copies vary with demand and prevents overloading and improves response time.

5.2.3 Node additions

In order to join the network, a new node creates its own public-private key pair. This uniquely identifies the node. The public key together with its physical address is then sent out to the network with a user-specified TTL. The receiving node notes the new node's information and then forwards it to another node chosen randomly from its routing table. Once the TTL becomes zero, all the nodes in this chain decides on a random GUID to be assigned to the new node in the keyspace using a cryptographic protocol and this is then notified to the new node and also added to the routing table of each node in the chain. Thus the key for the new node is allocated based on a collective agreement by all the other nodes in the chain.

As more requests are handled, the network's routing gets better. The routing tables should handle more clusters in order to improve the effectiveness of future queries. This is because the nodes get requests about keys that are similar to the keys it is associated with in other nodes' routing tables. Also, the data stores should handle more clusters of similar keys as the requests follow the same path as the inserts.

6 Security

Anonymity has been the central goal of Freenet project. This includes the protection of the requestors' and inserters' identities. Other security goals are: to protect the files against malicious modifications and denial-of-service attacks. Rubin et al.[17] in their work describe the taxonomy of anonymous communication properties. It is represented in three axes. They are:

- Type of anonymity

This means both sender and receiver anonymity. i.e., an attacker wouldn't know who created the message and to whom is it sent. In Freenet's case, since the keys identify the receiver, this would mean key anonymity.

- Adversary

This is the attacker or a malicious node.

- Degree of anonymity

This would range from absolute privacy (the communication cannot be perceived) to beyond suspicion (all nodes are equally probable to send and receive messages) and exposed.

Based on this taxonomy, the anonymity properties of Freenet are illustrated in Table 1.

System	Attacker	Sender Anony-mity	Key anonymity
Basic Freenet	Local eaves-dropper	Exposed	Exposed
	Collaborating nodes	Beyond Suspicion	Exposed
Freenet + pre-routing	Local eaves-dropper	Exposed	Beyond Suspicion
	Collaborating nodes	Beyond Suspicion	Exposed

Table 1 Anonymity properties of Freenet

7 Performance

Freenet shows better performance based on the simulations conducted. Some of those results are illustrated in figures 3, 4, 5 and 6 [15]. An extended version of results is provided by [18]. Simulations were conducted to test the scalability and fault-tolerance. The performance analysis of Freenet can be explained based on the small-world network model [19]. In this model, a majority of nodes will have fewer connections to other nodes while a smaller set of nodes will have a wider connection of nodes. Small-world networks are represented by power-law distribution of graph degree. In the case of Freenet, the graph degree is the number of routing table entries. Freenet shows the characteristics of the small-world networks as is shown in figure 3. It shows the graph degree distribution for a 10,000-node network simulation. The maximum of routing table entries here is 250. The small-world network enables shorter paths and greater fault-tolerance.

7.1 Scalability

The result of the simulation for analysing the scalability of Freenet is depicted in figure 4. The test started with 20 nodes. After every five inserts and requests (with TTL=20), a new node was added (with a TTL=10) and after every 100 inserts and requests, the network's performance was measured by sending a set of requests and

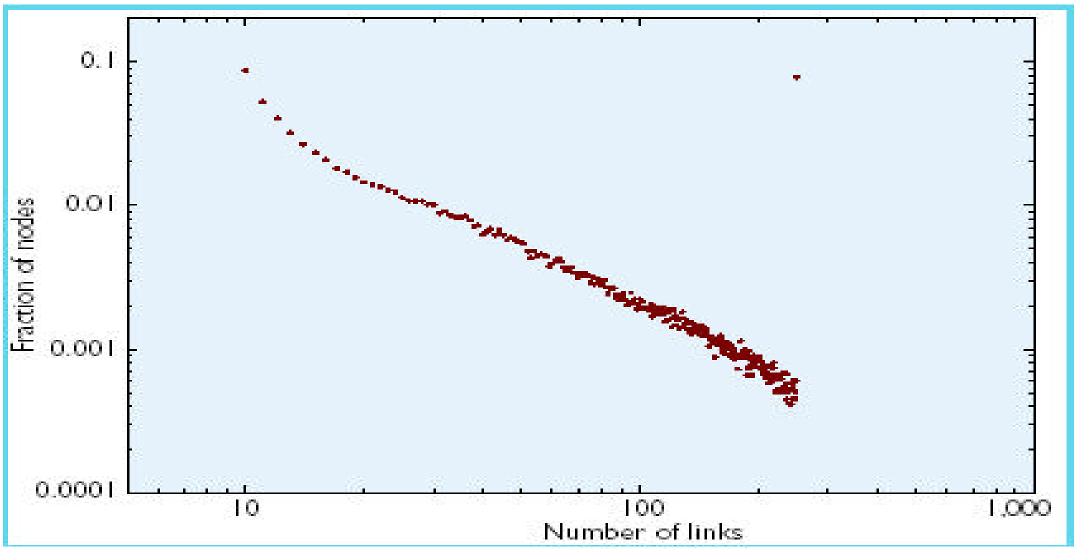


Figure 3 Degree distribution among Freenet nodes

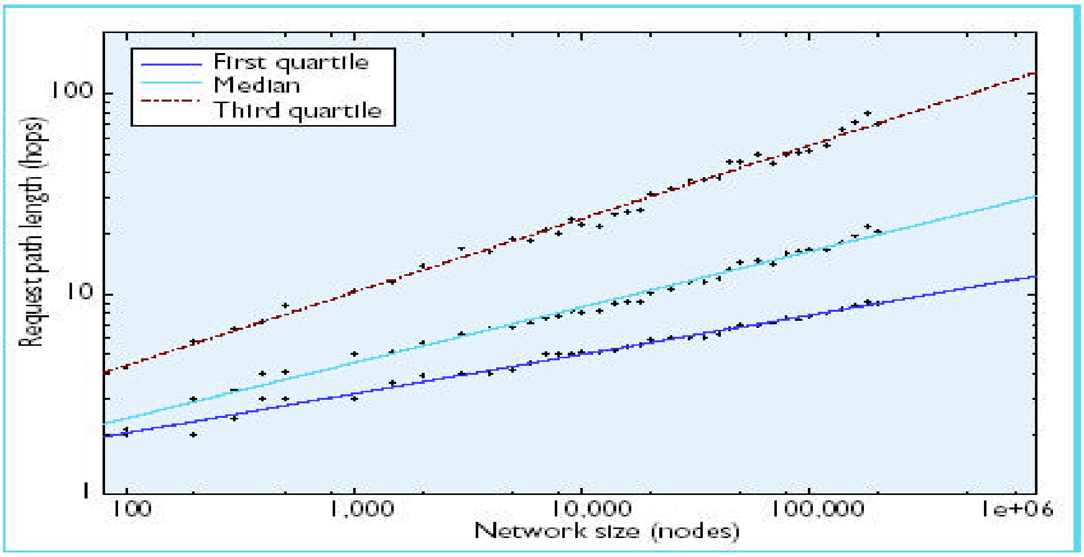


Figure 4 Request path length vs. Network size

recording the path length distribution. The test was conducted until the network grew to 200,000 nodes. The extrapolation of the results in Figure 4 shows that Freenet is capable of scaling up to one million nodes with a median path length of 30. The results in the figure are averaged over 10 trials.

7.2 Fault Tolerance

Tests were conducted to analyse the fault tolerance capabilities of Freenet under the cases of random failure and targeted attacks. Figure 5 shows the results of the simulation in the case of

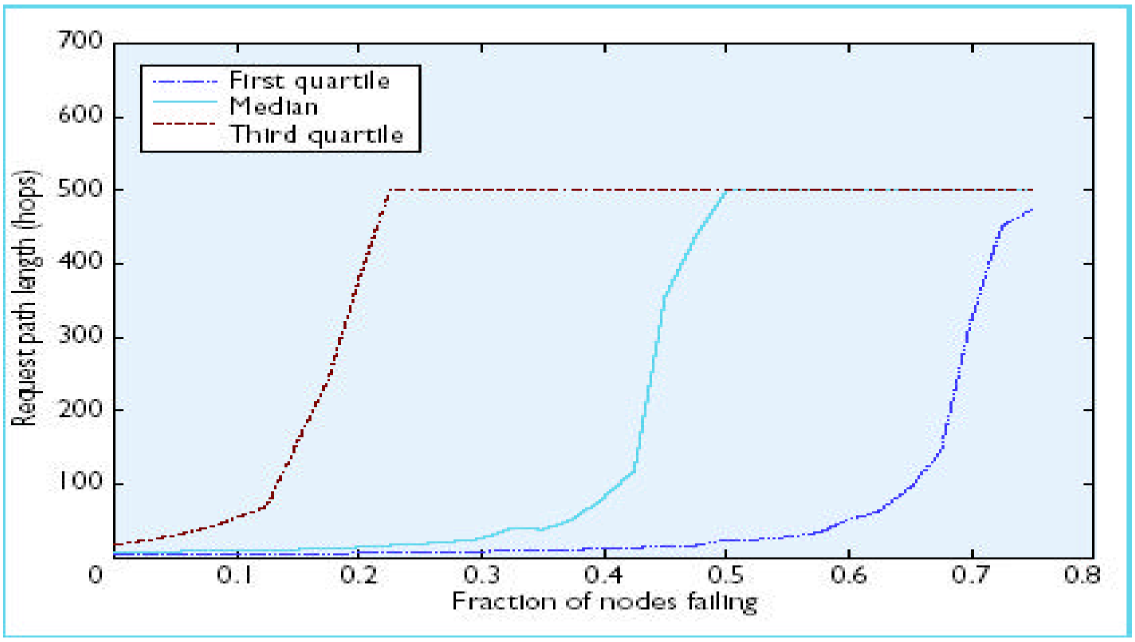


Figure 5 Request Path Length: Random failure case

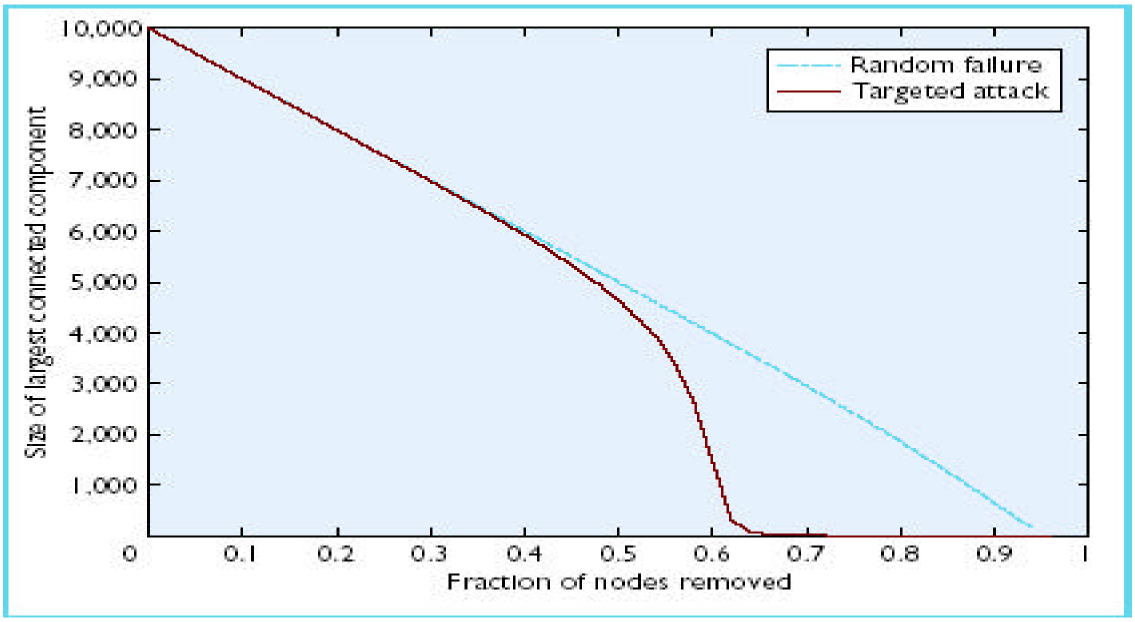


Figure 6 Connectivity: Random failure and targeted attack cases

random failures. Nodes were removed in random from a network of 10,000 nodes.

Results show that the path length remained below 20 even at 30% node failure showing the benefit of small-world network behaviour.

Figure 6 shows the size of largest connected component as the attacks were made randomly followed by targeted attacks. After 60% of node failures, the network drastically broke up into fragments as shown in the figure thus exposing some weakness in case of targeted attacks.

8 Work in Progress

Work is currently ongoing to develop the Next Generation Routing mechanism [20] for Freenet. The aim of this effort is to make the existing routing mechanism smarter by taking into consideration response time for requesting a certain key, the percentage of requests that succeeded in finding information and the time taken to establish a connection. This additional information would be added to a node's routing table and in the event of a new request being received, the information would be used in deciding the next node to which the request has to be forwarded.

The benefits of the Next Generation Routing is perceived to include capability in adapting to network topology (currently all nodes are treated equal and underlying network topology is ignored), ability to evaluate the performance of routing locally (mainly for faster development improvements) and optimisation of routes.

9 Usability Aspects

The author had the opportunity to use the Freenet application for the first time during the preparation of this paper. However, the experience was far from satisfactory, especially what comes to the speed of accessing the files. Freenet is a java-based application that helps to maintain platform independence. However, the systems without a java runtime environment (installing it sometimes require admin permissions as in the Networking Lab) may not be able to install Freenet. An operating system based Freenet application could have avoided this issue.

Some of the observations while using the application are:

- The response times for accessing Freenet sites were quite long.
- Too many "network busy" messages were experienced initially.
- The experience of browsing and response from the network gets better with continuous usage. This perhaps is due to the learning nature of the routing table of the node.
- A user-friendly search scheme is missing and hence the search for documents is an unpleasant experience. This might limit Freenet to be used only by geeks, thus restricting its development.
- The user experience to browse and search has been slightly improved using the web client interface Fproxy which comes with the basic application. This was useful to a greater extent as it provided some userf-friendly Freenet sites and links. Freenet also has clients that provide command line interface (CLI) features. A variety of other tools can also be found at [21].
- Freenet carries almost all types of file formats and content. This is evident from the categories available on one of the Freenet directory sites YoYo. These categories include news, music, literature, humour, blogs, philosophy, technology, video and adult.

Overall, there is a greater need and urgency to improve the user interface, provide search features and faster response time, if Freenet aims to achieve greater acceptance similar to other P2P applications like Kazaa among the non-technology oriented user groups. This would only enable the achievement of the main goals of the Freenet project.

10 Inference

Freenet is a scalable and fault-tolerant publishing system that has decentralized network architecture based on the P2P concept.

It has a packet-oriented protocol with self-contained messages. The major aim of this system is to enable freedom of expression online by maintaining the anonymity of all the participants involved in publishing or reading the information on the network. Freenet provides an efficient way for virtual global information storage.

While Freenet has come a long way since its inception in 1999, more work needs to be done in the area of denial-of-service attacks flooding the system with junk data. Also, absolute privacy needs to be provided by implementing a mechanism for key anonymity. Work is ongoing on caching policies, routing algorithm and also simulations and modelling aspect of Freenet. Freenet also needs to implement accountability features like trust and micropayment mechanisms as has been implemented by the Free Haven project. While Free Haven doesn't consider efficiency as a priority goal, Freenet is working towards that aim as well. With the inclusion of additional features similar to those currently available in the Free Haven project, Freenet can well become one of the best publishing systems online. The search mechanism also needs to be standardised. User experience needs to be enhanced by providing better and faster access to the content.

In future, it will be interesting to see how Freenet evolves and furthers the cause of free expression online. The routing algorithm of Freenet, i.e., steepest-ascent hill-climbing search, in particular would be useful in ad hoc routing.

11 References

[1] R. Schollmeier, "A definition of Peer-toPeer Networking for the classification of Peer-to-Peer Architectures and Applications", http://csdl.computer.org/comp/proceedings/p2p/2001/1503/00/15030101.pdf

[2] The Freenet Project, www.freenetproject.org

[3] D.L. Chaum, “Untraceable electronic mail, return addresses, and digital pseudonym,” Communications of ACM 24(2), 84-88, 1981.

[4] L.Cottrell, “Frequently asked questions about MixMaster remailers,” http://www.obscura.com/~loki/remailer/mixmaster-faq.html

[5] D. Goldschlag, M.Reed, and P.Syverson, “ Onion routing for anonymous and private Internet connections,” Communications of the ACM 42(2), 39-41, 1999.

[6] Anonymizer, http://www.anonymizer.com/

[7] M.Waldman, A.D. Rubin, and L.F.Cranor, “ Publius: a robust, tamper-evident, censorship-resistant, web publishing system,” in proceedings of the Ninth USENIX Security Symposium, Denver, CO, USA, 2000.

[8] The Free Haven Project, http://www.freehaven.net/

[9] Distributed.net: Node Zero, http://www.distributed.net/

[10] Napster.com, http://www.distributed.net/

[11] Gnutella, http://gnutella.wego.com/

[12] Akamai, www.akamai.com

[13] I. Clarke, O. Sandberg, B. Wiley, T.W. Hong, “ Freenet: A Distributed Anonymous Information Storage and Retrieval System”, Lecture notes in Computer Science 2009, H.Federrath, ed., Springer-Verlag, Berlin, 2001, pp. 46-66.

[14] Freenet source code and installation details, http://www.freenetproject.org/index.php?page=download

[15] I. Clarke, S.G.Miller, O. Sandberg, B. Wiley, T.W. Hong, “Protecting Free Expressions Online with Freenet”, IEEE Internet Computing, January-February 2002.

[16] ANSI, ANS X9.30.2-1997: Public Key Cryptography for the Financial Services Industry-Part 2: The Secure Hash Algorithm (SHA-1).

[17] M.K.Reiter and A.D. Rubin, “ Anonymous web transactions with Crowds,” Communications of the ACM 42(2), 32-38, 1999

[18] T.Hong, “Performance,” in Peer-to-Peer: Harnessing the Power of Disruptive Technologies, A.Oram, ed., O’Reilly and Assoc., Sebastopol, Calif., 2001, pp.203-241.

[19] D.Watts and S.Strogatz, “Collective Dynamics of ‘Small-World’ Networks,” Nature, no.406, July 2000, pp.378-381.

[20] Freenet ‘s Next Generation Routing Protocol, http://www.freenetproject.org/index.php?page=ngrouting

[21] Freenet Tools, http://www.freenetproject.org/index.php?page=tools

Peer-to-Peer Communication Services in the Internet

Markus Isomäki
Nokia Research Center
Itämerenkatu 11-13, 00180 Helsinki
markus.isomaki@nokia.com

Abstract

This paper explains in generic terms how peer-to-peer technology works in today's Internet. The technical issues related to network self-organization, searches and network address translator (NAT) traversal are discussed. It also explains how IP communication services, such as voice over IP (VoIP), work based on traditional approaches, such as SIP. A new peer-to-peer approach to provide this kind of services is analyzed by making a case study on a peer-to-peer VoIP software called Skype, which was introduced in the Internet in September 2003. The goal of this paper is not to explain any particular technology or protocol in detail, but rather give more general overview on the topic of peer-to-peer technology applied to IP communication services.

1 Introduction

1.1 Peer-to-Peer Phenomenon in the Internet

Peer-to-peer applications have broadened the way in which Internet is used quite dramatically. Still at the end of 1990's the Internet traffic patterns were dominated by HTTP traffic resulting from WWW usage. At some point it was reported that around 95% of Internet traffic was HTTP.

Within the last three years the situation has changed. The absolute amount of HTTP traffic has still grown, but peer-to-peer traffic has in many places surpassed it in volume. Recent estimates mention the share of peer-to-peer traffic out of the overall Internet backbone traffic volume to be somewhere between 40% and 60%. The total number of users is in the order of hundreds of millions.

The main application for peer-to-peer protocols in the Internet has so far been sharing of files, mostly music and movies. There are several different (albeit quite similar in terms of technology) protocols or systems for this, and their success has clearly shown their power. Obviously part of the success is based on the fact that peer-to-peer is an excellent way to circumvent the copyright laws and distribute illegal material. But even from purely technical point of view peer-to-peer file sharing and searches have major advantages over traditional mechanisms such as WWW/HTTP, WWW search engines, WebDav etc., which rely on a more centralized content sharing paradigm.

1.2 Peer-to-Peer Entering the IP Communication Services Space

Another, longer term, trend in the Internet has been the emergence of the so called IP based Communication Services, such as voice, video, instant messaging and presence. VoIP has started to gradually happen since around 1995, and by this date already millions of people have used it, at least just for fun. It is expected that this trend will continue, and within the next ten years already a large proportion of the voice traffic in developed countries would be carried in IP-based networks.

In the recent years IP Communication Services have been driven by standards from the Internet Engineering Task Force (IETF). The main protocol in this space is the Session Initiation Protocol (SIP), which is able to initiate peer-to-peer media sessions, but also relies on traditional client-server role separation in some of its functionalities, and is thus quite different from today's peer-to-peer file sharing technologies.

Until September 2003 there was not much talk about applying the true peer-to-peer technology to Communication Services. However, at that date an application called Skype was released. Skype uses some of the very same principles as the peer-to-peer file sharing systems. The main difference is that this time it is people (or more specifically, user IDs) instead of files that are searched, and after a successful search real time communication (such as a voice conversation) takes place rather than transfer of a file.

1.3 Organization of This Paper

The rest of this paper is organized as follows. Section 2 is an introduction to the world of peer-to-peer applications and protocols in general from various angles. Section 3 discusses some of the basic technologies behind the current peer-to-peer systems, as far as it is known, as many of the actual protocols are still not opened even by reverse engineering. Section 4 then explains the basic aspects of IP Communication Services, and compares the predominant approaches (mainly SIP, but the same applies to other protocols such as H.323 too) to the pure peer-to-peer paradigm.

Section 5 is an introduction to Skype, explaining some of the technologies it is based on. Unfortunately, it seems that at this date it is not possible to find any accurate definitions on Skype protocol details by WWW searches let alone from more official literature, and reverse engineering the protocols from scratch would be far beyond the intended scope of this paper. Thus, many details still remain a mystery, even if the general aspects can be somewhat understood.

Section 6 finally makes some short conclusion and points areas for future research, for which there definitely is room in this novel area.

2 Peer-to-Peer in Theory and Practice

2.1 Definition of Peer-to-Peer Protocols

There are many definitions on what peer-to-peer protocols or systems mean.

One of the best definitions from protocol engineering point of view is given by the Internet Research Task Force (IRTF), that has an active research group on peer-to-peer protocols:

“Peer-to-Peer (P2P) is a way of structuring distributed applications such that the individual nodes have symmetric roles. Rather than being divided into clients and servers each with quite distinct roles (such as Web clients vs. Web servers), in P2P applications a node may act as both a client and a server. P2P systems are in general deployable in an ad-hoc fashion, without requiring centralized management or control. They can be highly autonomous, and can lend themselves to anonymity. “ [1]

“A key concept for P2P systems is to permit any two peers to communicate with one another in such a way that either ought to be able to initiate the contact. As such, P2P is a powerful tool for organizing cooperative communities - both in the research and commercial domains - with common goals.” [1]

There are also a couple of good examples of IETF-standardized protocols that are widely used in the Internet, and that fit to this description: NNTP, the protocol used by USENET (Internet news) servers to exchange newsgroup messages; and BGP, the protocol used by Internet core routers to exchange routing database information.

2.2 Peer-to-Peer from Deployment and Operation Point of View

It is possible to define peer-to-peer also by looking at the model how these applications currently work from deployment and operation point of view. In this aspect the peer-to-peer applications could be classified as self-organizing or self-sufficient. All they need to work is IP connectivity.

In most systems there are no official operators or service providers, let alone federations or agreement between providers. There are typically some fixed servers to help new nodes in bootstrapping, beyond that no maintenance or personnel is needed. The application level infrastructure is instead dynamic and is established by co-operation of the application instances that the normal end-users install and run in their personal PCs.

In a recent interview the developers of KaZaA and Skype defined peer-to-peer as follows: “Software is not peer-to-peer just because it establishes direct connections between two users; most Internet software does this to some extent. True P2P software creates a network through which all clients join together dynamically to help each other route traffic and store information.” [2]

In suitable circumstances (fast network connectivity, public IP address in use, long running time) the application instances can dynamically assume the role of a “supernode” (the terminology differs among systems, but the general idea is the same). In principle any node can become a supernode. From ordinary node’s point of view the supernodes form the server infrastructure that in ordinary systems is fixed. In some peer-to-peer systems there is yet another layer of hierarchy to optimize the dynamic topology and communication.

The supernodes communicate with each other in order to form an application layer network that will forward requests and do other operations on behalf of ordinary nodes. In a typical case each ordinary node is aware or connected to at least one supernode, and the connections between the supernodes aim to form a fully connected graph, so that basically it is possible to connect any nodes with each other through some route within the graph. It should be emphasized that the supernodes are only used to locate the other nodes (peers), and the actual application data transfer always happens directly between the originator and the target node, peer-to-peer.

It is reported that in some systems the ratio between supernodes and ordinary nodes is around 1:100, but this probably varies a lot.

From user's point of view this means that it is possible simply to download the application from the WWW, install it and start running. If the user's computer is connected to the Internet via a broadband connection and possesses a globally routable IP address, it is possible that his application actually starts running as a supernode, even without the user knowing about this.

The main observation from Internet topology point of view is that the nodes providing the actual service are thus mainly located in ordinary Internet users' homes, rather than in server hotels, where majority of traditional WWW, e-mail or FTP servers are run and are hosted. There is also no authority assigned with a large portion of the infrastructure, since everything is distributed among hundreds or thousands or millions of individuals. This makes the attempts to control the systems very challenging. Also, like any self-organizing network topology, it is very resilient against failures in parts of the system.

It is clear that the growth of the broadband Intenet access market has been the pre-requisite to the success of peer-to-peer systems, but nowadays the situation seems to be even vice versa!

2.3 Applications in Use Today

In practice majority of the peer-to-peer applications that are used in the Internet currently are meant for search and transfer of files. Most of the content shared in the peer-to-peer systems is commonly believed to be pirate music in MP3 media format and pirate movies in DIVX media format, even if there is no formal statistical evidence of this.

Typically the protocols used are not public, although some specifications established through reverse engineering can be found through WWW searches.

The most famous peer-to-peer file sharing systems in use today are called Gnutella, KaZaA, eDonkey and Direct Connect. Gnutella protocol specification can be found easily in WWW, but a proper description of the others is harder to find. It seems to be however so, that Gnutella is a good starting point, since it is peer-to-peer in its most classic and simplistic form. The other systems, such as KaZaA (which is important since Skype is based on it) borrow a lot from Gnutella. They then add some optimizations on top of it, for good or worse. The basic principles in all still seem to be pretty similar.

2.4 Known Problems in Today's Peer-to-Peer Applications

The practical reality of peer-to-peer systems in the Internet is not without major issues. The main problems associated with many of today's peer-to-peer applications are:

- Spyware: At least according to the common wisdom many Peer-to-Peer applications are plagued with hidden code that has nothing to do with the actual purpose of the application. This is then used to e.g. send advertisements to the user, or report his WWW browsing habits to some mysterious parties without user's concent.
- Viruses: Peer-to-peer file sharing is an ideal way to spread viruses on the Internet.
- Breaking security policies: As explained in Section 3.4, Peer-to-Peer applications know ways how to traverse firewalls, which is not the intention of the people who have installed such devices in the network.
- Stealing bandwidth: Peer-to-Peer applications often do not care for TCP-like congestion control, but try to grab more bandwidth. This is nice for the individual using such an application, but a catastrophe in congested networks for users having some well-behaving clients. The situation is probably worst in places where a relatively small number of users is sharing the Internet access connection. Peer-to-peer traffic can there steal the bandwidth from all the other applications.

3 Technology Behind the Peer-to-Peer Applications

There are four interesting technical issues related to how the most common Peer-to-Peer applications used today really work:

1. Allocation and discovery of supernodes
2. Performing the searches (queries)
3. NAT traversal
4. Firewall traversal

3.1 Allocation and Discovery of Supernodes

Not much is documented in general about the supernode allocation techniques. Even the Gnutella protocol specification does not contain any explanation on this. However, it is clear that having public IP address, fast network connectivity, short "ping" roundtrip times to other supernodes and long running time of the host (no shutdowns or network interrupts) are the main factors the allocation algorithm takes as input.

How bootstrapping nodes discover supernodes is also not documented in detail. By doing traffic analysis it is possible to deduce that when the application is first time installed and run, it typically uses some kind of hard coded list of supernode addresses to make the initial contact. This means that at least some of these “seed” supernodes need to be always running and thus really operated by someone, otherwise the system would fail.

However, when the application is first time able to connect to any of the “seed” supernodes, it gets a list of some other currently running supernodes, which is used and kept updated from that point on. Thus, when the user starts the application the second time, it may no longer need to rely on any static configuration.

3.2 Performing Searches

Peer-to-peer systems give the impression, that it is possible to make searches/queries that cover the content available in the whole network of nodes. How this works in detail depends on the system, but Gnutella can be used here as a basic reference. The specification of Gnutella can be found in [3]. Searches can be made typically based on a large set of metadata, such as file name, format, artist name, bitrate etc., obviously depending what type of content is searched.

In Gnutella the QUERY descriptors (messages) carry the search information, and a node always sends them to all supernodes it is connected to. The supernodes in their turn forward the QUERY to all ordinary nodes or supernodes they are connected to, unless they have seen a QUERY with the same descriptor identifier already, in which case they discard it. Another reason to discard a QUERY is that the time-to-live value, which is decremented by one by at each hop, reaches zero. In this way the QUERY will propagate within the graph consisting of supernodes as shown in the figure below. (If TTL is small, obviously the search covers only a subset of the graph.)

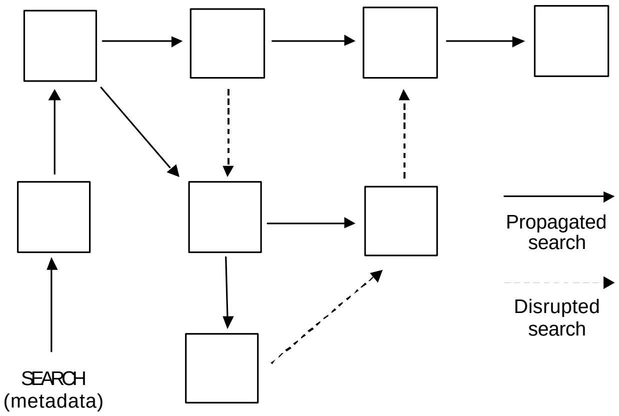


Figure 1. Searches in Peer-to-Peer Systems

Whenever a node or a supernode having the content that matches the QUERY is reached, the node will send a response back toward the originator of the query. The response will contain the IP address and the TCP port number where the content can be obtained from the responding node. Supernodes are also able to cache this information, so the leaf nodes do not need to be queried every time. (In any case it is likely that several responses are generated, as the same content can be found from multiple locations).

After receiving a suitable response the originator can open a direct a TCP connection to the address received, and fetch the desired content, as shown in the figure below. This is the essence of the Peer-to-Peer applications: The content can reside on any host running the application (either in ordinary or supernode mode), and content fetching is done directly from there. For this Gnutella uses HTTP, but other protocols are certainly also possible. (In practice the content is often downloaded from multiple places simultaneously, so that each download provides a different part of the whole. This makes the overall download times shorter, but also congests the network, as it means multiple simultaneous TCP connections.)

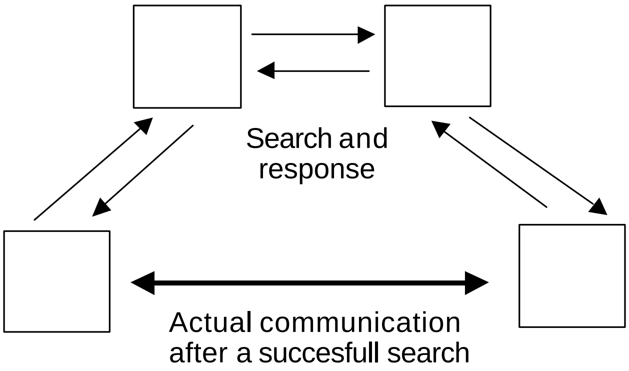


Figure 2. Direct peer-to-peer communication after successful search

Based on the information on Gnutella it can be concluded that these searches are not very well optimized, as there is no indexing. However, they are exhaustive and most importantly have been demonstrated to work in practice too. More advanced systems than Gnutella have their ways to improve the search efficiency.

3.3 NAT Traversal

A large number of IP hosts in today’s IPv4 based Internet are behind NATs. This means that any application or protocol that wishes to be widely deployed must have ways to traverse them. This is especially important for peer-to-peer applications, where two hosts that are behind their respective NATs need somehow to be able to communicate.

Current peer-to-peer protocols utilize a few well known techniques to establish communication through NATs. In some of these the supernodes, which have public IP addresses, are used. A good description of these techniques can be found in [4].

The suitable technique depends on two main issues:

1. What kind of NAT is being traversed
2. Is the traffic that needs to traverse the NAT transported on top of TCP or UDP

3.3.1 TCP communication

In most cases TCP is more problematic, since NATs typically only allow TCP connections opened from "inside". In this case the two mostly used options are relaying and connection reversal.

If both communicating hosts are behind a NAT, relaying is usually the only viable method. The basic principle of relaying is shown in the figure below. There needs to be an IP host with a public IP address that can act as a relay between the communicating parties. Both parties open a TCP connection to the relay, and the relay will then forward all traffic from one connection to the other. Typically each host behind a NAT needs to have a relay associated with it a priori to a communication request. In that way the relay's address can be given to the communicating party instead of the actual address owned by the host using the relay. IETF's TURN protocol [5] is one good example how relays are allocated and used. In Peer-to-Peer applications, supernodes act as relays to help the nodes behind NATs to communicate. Thus, relay allocation is as dynamic as is the allocation of supernodes.

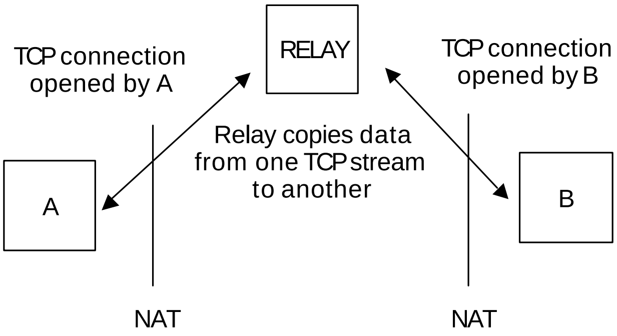

TCP connection opened by A
RELAY
TCP connection opened by B
Relay copies data from one TCP stream to another
A
B
NAT
NAT

Figure 3. Using a relay to traverse NATs

There are obvious reliability and delay problems introduced by the relays. In worst case scenario two parties behind a NAT in Finland are allocated a relay in Australia, running on a PC at someone's home. Traffic flow is slow, and in the middle of the communication, the user in Australia can shut down his computer, killing the connection. These problems can be eased by clever supernode/relay allocation algorithms, but the situation is never perfect in pure dynamic Peer-to-Peer environment.

The problem is eased if only one of the communicating hosts is behind a NAT. This host still can not be reached via a direct TCP connection from outside, but instead a relay can be used just to tell the host to open a connection to the host that wishes the communication to be established. After that the relay does not play any role in the communication. This scenario is depicted in Figure 4. For instance Gnutella supports this mechanism through the PUSH descriptor (message), which commands the host behind the NAT to open a connection to the address provided within the PUSH descriptor.

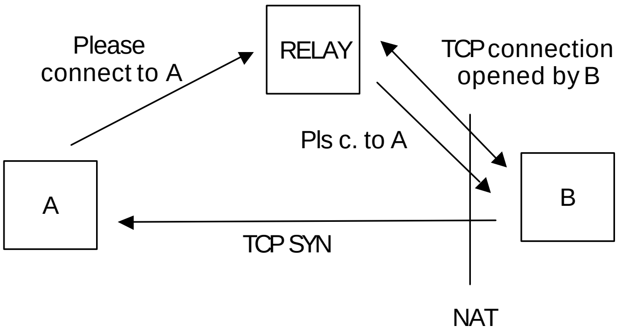


Figure 4. Connection reversal technique

In both of the above mechanisms some keep-alive signaling is needed periodically to keep the NAT binding for the TCP connection active. The binding expiration times vary greatly, so optimizing this is not easy. The problem is not relevant if the TCP connection is constantly used anyway, e.g. for file transfer.

3.3.2 UDP communication

In case of UDP the type of the NAT plays an important role. A UDP packet sent through the NAT from "inside" to "outside" always creates a binding within the NAT between the internal and the external address/port pairs. Some NATs forward packets sent from any address to the external (public) address according to an active binding. However, some NATs only allow packets from the address where the original UDP packet that established the binding was sent to.

In either case many peer-to-peer applications know how to utilize this "UDP hole punching" technique. A host behind a NAT can use a supernode with a public IP address to find out its own public address given by the NAT. This public address can then be given to parties which need to initiate connections to the host behind the NAT. IETF's STUN protocol [6] is one example of this kind of mechanism.

The main problem, as with TCP, is the need to keep the binding alive by sending packets through the NAT periodically.

3.4 Firewall Traversal

While NAT traversal is merely a way to cope with current Internet's deficiencies, firewall traversal can be defined as violation of security policy. This of course depends whether the authority in charge of the firewall approves the traversal or not.

Without asking such permissions, some Peer-to-Peer applications can traverse firewalls as well as NATs. In some cases the same relaying and connection reversal mechanism that are used with NATs work equally well with firewalls. However, especially many corporations have a security policy that allows only HTTP traffic to or from some proxy server to pass through the firewall. To cope with this kind of strict policy, the peer-to-peer protocols can tunnel themselves within fake HTTP/TCP connections. There is almost no way to block this, unless the firewall is able to understand the actual content of the packets. Note that while at least some HTTP headers need to be carried on top of TCP to get through the HTTP proxy, it does not mean that the protocol has to follow the rules of HTTP beyond this. This means that e.g. real time traffic can be carried this way more efficiently than what using HTTP would normally imply.

4 IP Communication Services

In this paper we use the term "IP Communication Services" to cover a wide range of person-to-person communication means, such as voice and video communication and instant messaging. Also the presence service, which allows people to see other people's ability and willingness to communicate with these means, is included.

The main functions that need to be performed by any full-blown IP Communication Services protocols are:

- Locating the parties to be involved in the communication based on some identifier that represents these parties,
- Negotiating the addresses and parameters needed for transferring the actual media (such as voice) involved in the communication session,
- Carrying the media.

In most cases the systems implementing communication services exhibit some sort of peer-to-peer behavior. Endpoints need to be able to both initiate and receive communication sessions, and the most efficient way to carry the actual media is end-to-end between the communicating endpoints, not via network-based servers. The "traditional" protocols, however, also have some significant differences compared to the pure peer-to-peer approach discussed in Sections 2 and 3.

4.1 Traditional approach to Communication Services – Session Initiation Protocol

IETF's Session Initiation Protocol (SIP) [7] is a canonical example of an IP-based communication services signaling protocol. It allows users to initiate voice and video calls (or in general any type of end-to-end sessions), send instant messages to each other and monitor each other's presence information.

SIP uses several methods normally associated with peer-to-peer applications. For instance servers are used only to locate other endpoints (User Agents) and after that both signaling and the actual media can be sent directly between the communicating endpoints.

On the other hand SIP differs from the pure peer-to-peer paradigm a la Gnutella in that it makes a clear distinction between client and server protocol entities. Endpoints are called User Agents, and they rely on fixed proxy and registrar servers, and these roles are not meant to be determined dynamically.

There is also a clear and fixed hierarchy established by DNS domain names. Each domain that runs SIP service must arrange independently from any other domain its Registar and Proxy services, and if those are down, there is no way for other domains to compensate this. The assumption is that each domain has some kind of operator or administrator, who takes care of these servers. It is surely possible for anyone to run their own Registrar/Proxy service at home provided that a DNS domain is also obtained for this purpose. In this case the user just takes the role of the operator himself, but this does not change the organization of the protocol entities.

Figure 5 below depicts a typical SIP communication process. Each user has a unique SIP URI allocated from his home domain, and if the user wants to be reachable, he must be registered to his home domain through some device running a SIP User Agent. In the example shown in the figure UserB is registered to his home domain domainB in Step 1. In Step 2. UserA who is located at another domain (domainA) wishes to initiate communication with UserB@domainB. His User Agent thus issues a SIP protocol request to his own serving SIP Proxy, who then searches the DNS database for SIP Proxy service in domainB. (It is possible to omit this step if the User Agent can make the DNS search itself.)

After a successful search the SIP request can be issued to domainB's Proxy, who can forward it to UserB's User Agent based on the registration information. After UserB has replied to UserA, the User Agents can start direct communication with each other, and also the media streams can be sent directly between the IP addresses negotiated within the signaling exchange. Also it is possible to send any further signaling (such as adding a new media) directly between the User Agents

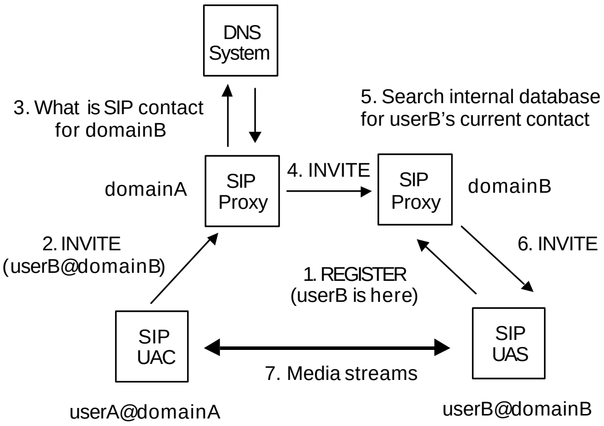

DNS System
3. What is SIP contact for domainB
5. Search internal database for userB's current contact
domainA
SIP Proxy
4. INVITE
SIP Proxy
domainB
2. INVITE (userB@domainB)
1. REGISTER (userB is here)
6. INVITE
SIP UAC
SIP UAS
7. Media streams
userA@domainA
userB@domainB

Figure 5. Locating other users and setting up media connections using SIP.

SIP's capability to locate other users is very scalable, and the requests can be routed to correct locations very efficiently making it a suitable signaling protocol even for services which require relatively fast communication establishment, such as telephony.

The weaknesses of SIP compared to pure peer-to-peer model is the reliance on fixed server infrastructure, which makes its deployment tedious. In practice some personnel are needed to operate this infra. Also most SIP implementations currently have no interoperable ways to traverse NATs or firewalls, even if IETF's STUN and TURN protocols have been developed mainly for this purpose. As SIP is a standard assumed to be implemented by various independent vendors, full interoperability also requires more testing than what is so far done.

The current situation with SIP is that there are several single-operator single-vendor islands offering SIP services, but the inter-operator or multi-vendor solutions are still very rare, and need to be designed case by case.

5 Skype – The Peer-to-Peer Communication Service Solution

5.1 A Short History of Skype

Skype is the brainchild of the developers of KaZaA, lead by Janus Friis and Niklas Zennström, residing in Sweden. The software was launched to public in September 2003, and as of November 20th 2003 has been downloaded already more than 2.8 million times. (The claim is that there has been already over 100 000 simultaneous users in the system.) It is available for free from http://www.skype.com, and it runs on the Windows platform only. The most recent software version on this date is Beta 0.95.

According to the developers there are plans to provide gateway services between Skype and PSTN users, as well as between Skype and SIP users in the future.

The developers do not intend to give out any real details how their protocol works. It is actually even forbidden to monitor Skype network traffic according to the license agreement that the user has to accept when installing Skype! For this reason it is not probably even legal to publish any information about the protocol that can only be obtained in this way. At the time of writing this paper the author was not able to find (through WWW search engines) any proper technical documentation of Skype's protocols.

The Skype website provides this kind of information on the future plans:

"During the beta period Skype is free and helps us to refine and improve our product. Eventually, some features and services of Skype will require a paid subscription or prepayment. Our ambition is to keep the basic functionality of Skype (PC to PC calls) free. More information will be provided once our beta program is complete." [8]

5.2 Features Provided by Skype

Skype allows users to make Voice over IP telephony calls and send instant messages between each other, as well as monitor the presence information of other users by the usual concept of a "buddy list". Users can also search other users based on some metadata such as language or city of residence (the user fills in this information when he installs Skype, and the system maintains this information for three days after the user's last login). Each user is identified by a Skype user identifier, which is an unstructured text string, and which the user is able to select for himself, provided that it is still free.

Skype works through NATs and Firewalls, provided that outgoing TCP traffic to port 80 is allowed. It does not work in environments, where the only allowed traffic is HTTP via a proxy.

If the underlying IP network has enough capacity, Skype provides better voice quality than PSTN by using voice codecs that reach up to 7 kHz audio frequency (compared to 3.4 kHz in PSTN/PCM). If the network connection is slow (modem or ISDN), the voice quality is also worse due to the use of lower bitrate codec. Skype claims to be able to adapt to whatever the network conditions at a particular moment are.

Skype encrypts both singaling and voice traffic so that confidentiality of the transferred information is achieved.

5.3 User Experience

The author used Skype version 0.93 a few times during early November 2003. Here are some subjective notions based on that use:

- The software was very easy to install and use.
- The searches for other users work very fast.
- Sound quality in fast network (Ethernet and WLAN were used) is excellent, and clearly better than in PSTN.
- Call setups are very fast if the other user is on your "buddy list". If not, it can take a while.
- The updates of presence information are rather slow. It can take several minutes before your application notices that the other user is no longer available. Also, if you try to call to a user who your application thinks is available and who actually no longer is, it will take quite long before the application gives up. This is probably because the failure to connect is first interpreted as an issue with NAT or firewall, and the software then tries out several methods to establish a connection.
- As expected, Skype did not work through a corporate firewall, which only allows HTTP traffic via a proxy to go through. Actually it was not possible to even register as Skype user from behind such a firewall.

5.4 The Technology

As no proper technical specification is available, it is hard to really provide any details on how Skype works. The following information is partly speculation, partly based on hearsay, partly on own usage and partly on what the developers have themselves claimed.

(When trying out Skype, the author did some extensive traffic and file I/O tracing by using Ethereal and Filemon, but it was nearly impossible to deduce anything from that beyond to which addresses the client was communicating and whether it was using TCP or UDP. From Filemon statistics it was possible to see that the software was reading Internet Explorer temporary files, but the reasons for this remain unknown.)

Skype is very much based on KaZaA's technology, which in its turn is very much based on Gnutella. The main requirement compared to KaZaA is that this time the searches have to be able to cover the whole network, as each search for a particular user identity will only result in one match at most, unlike in file searching, where there are usually several matches. To achieve this and to have faster search times it seems that some database synchronization among some high-level supernodes is perfomed. Skype website provides this kind of information.

"The Global Index technology is a multi-tiered network where supernodes communicate in such a way that every node in the network has full knowledge of all available users and resources with minimal latency." [8]

It may be that some kind of indexing is also used to distribute the information in a more intelligent manner than e.g. in Gnutella.

Because of the encryption it is hard to say anything about the protocol even if capturing the packets sent by Skype. The Skype website has the following information on the encryption used:

"Skype uses AES (Advanced Encryption Standard) - also known as Rijndel - which is also used by U.S. Government organizations to protect sensitive, information. Skype uses 256-bit encryption, which has a total of 1.1 x 1077 possible keys, in order to actively encrypt the data in each Skype call or instant message. Skype uses 1536 to 2048 bit RSA to negotiate symmetric AES keys. User public keys are certified by Skype server at login." [8]

When a Skype client is installed and first run it makes queries to some hard coded IP addresses, some of which are obviously servers run by the Skype developers or their associates. From those queries the client learns about the available supernodes. When tried several times, the client every time connected to a different set of supernodes, so the dynamic update clearly worked. Also encryption seemed to be in place, as it was not possible to find any plaintext information in the payloads, including search strings or IP addresses.

Call establishment signaling is run on top of TCP. If the callee is on caller's buddy list, TCP SYN is sent directly to callee's IP address, which means that the presence information contains also the direct contact information of each buddy. The actual voice traffic is carried on top of UDP (probably also RTP is used, but it is hard to tell due to the encryption). Skype supports supernode based relays to carry voice through NATs. Apparently the selection process for a topologically optimal relay is not so good, as the relays can be almost anywhere. Firewall traversal based on TCP port 80 usage is also supported, but as mentioned earlier, HTTP tunneling is not. If the Skype client for some reason fails to connect to a supernode, or call setup seems to fail, it tries to open connections to several places (at least 5-6 connection attempts).

Skype uses Global IP Sound (GIPS) voice codecs. These include a high quality codec used for fast connections, and a low bitrate codec used for modem/ISDN connections and in congested networks. [9] [10] GIPS codecs are royalty free, and have been from the start optimized for packet loss, i.e. they can cope with the loss of a large consequent block of bits, something which typically only happens in packet networks.

When a Skype client is started, it determines what kind of connectivity it has, i.e. LAN vs. dial-up, from Windows OS. Based on this and sending ping messages (not ICMP, but Skype's own pings), the client is also able to determine roughly how much capacity it has available in the network. The selection of offered codecs in call setup is determined based on these kind of heuristics.

6 Conclusions

Peer-to-peer applications have definitely been the hottest topic related to the Internet for the last few years. They have proved to work very well for file sharing.

The applicability of peer-to-peer systems to IP communication services, such as Voice over IP, is a novel concept, introduced first by Skype.

It is yet too early to say how successful Skype will be, but at least so far the technology seems to be working reasonably well. It is, however, unlikely that Skype or similar systems could, despite of their easy deployability, in the short term make such radical changes to the Internet usage as the peer-to-peer file sharing has done.

More traditional IP communication technologies, such as SIP, are also at the brink of deployment by operators. In the commercial space Skype, due to its proprietary nature, probably is not able to compete with them. It is hard to imagine how Skype could do all the functionality developed for SIP. However, in the "free usage" market segment Skype and similar applications are most likely going to become the winners.

7 References

[1] Internet Research Task Force: Peer-to-Peer research group charter, http://www.irtf.org/charters/p2prg.html

[2] News.com interview of Janus Friis, http://news.com.com/2008-1082_3-5074558.html

[3] Anonymous: The Gnutella Protocol Specification v0.4, Document Revision 1.2, http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf

[4] Ford, B.: Network address translation and peer-to-peer applications, Internet-Draft, April 2003 (work in progress), http://www.ietf.org/internet-drafts/draft-ford-natp2p-00.txt

[5] J. Rosenberg et al.: Traversal Using Relay NAT (TURN), Internet-Draft, October 2003 (work in progress), http://www.ietf.org/internet-drafts/draft-rosenberg-midcom-turn-03.txt

[6] J. Rosenberg et al., STUN - Simple Traversal of User Datagram Protocol (UDP) hrough Network Address Translators (NATs), IETF RFC 3489, ftp://ftp.rfc-editor.org/in-notes/rfc3489.txt

[7] J. Rosenberg et al.: Session Initiation Protocol, IETF RFC 3261, http://www.ietf.org/rfc/rfc3261.txt?number=3261

[8] Skype website, http://www.skype.com

[9] A. Duric et al.: Internet Low Bit Rate Codec, Internet-Draft, October 2003 (work in progress), http://www.ietf.org/internet-drafts/draft-ietf-avt-ilbc-03.txt

[10] Global IP Sound website, http://www.globalipsound.com

Protocols for resource management in a Peer-to-Peer Network

Yang Qiu
Networking laboratory
Helsinki University of Technology
yangqiu@cc.hut.fi

Abstract

Peer-to-peer computing is a term used to describe the current trend toward utilizing the full resources available within a widely distributed network of nodes. These resources include the exchange of information, processing cycles, cache storage, and disk storage for files. As most of the Peer-to-Peer networks overlay on the Internet, the resource management of the Peer-to-Peer network becomes an extremely important issue. First step toward a robust Peer-to-Peer network is the extension of centralized models of resource sharing (e.g. Napster) to a decentralized network system (e.g. Gnutella). After more recent attempts (e.g. CHORD and PASTRY), the limitations of such networks can be summed up in 5 issues, namely, performance, reliability, scalability, maintenance, and usability. There are 5 features that seem to capture these issues. These are Naming, Structuring, Locating & Routing, Data Managing and Topological Updating.

1 Introduction

The key idea of resource management is to share and access resources, especially in a dynamic changing environment of a P2P network. The announcing of its own resources and discovering resources provided by other peers is the mechanism of resource management, it contributes to the performance, scalability, maintenance, reliability and usability of the Peer-to-Peer Network. Nowadays, most of the Peer-to-Peer network systems align their peers in an overlay network to existing network infrastructures, mostly IP based Internet networks. The mechanism of resource management relies on its naming, structuring, locating & routing, data managing and Topological Updating methods.

Napster is a centralized network, and it works fine for less than 0.1 million subscribers. But if the amount of subscribers increases, the performance of the server will slow down. Gnutella is a broadcast decentralized network, and it also works fine for a small amount of subscribers. If the amount of subscribers increases, the workload of the peer will be too much to bear. Some approach is attempted, which keeps a decentralized hash table but splits it. So a search query will not be broadcasted to every peer. Chord and PASTRY are the protocols to realize this idea. Chord and PASTRY are not complete p2p file sharing network systems, but p2p applications can be built on top of their resource management mechanism.

In this paper, 4 networks are being investigated; they are Napster, Gnutella, Pastry and Chord. Napster and Gnutella are the 'ancient' and simple networks; Pastry and Chord create enhanced networking capability to the earlier Peer-to-Peer networks. So, in this paper, these 4 networks will be analyzed, evaluated and compared.

1.1 Evaluating the resource management of Peer-to-Peer Networks

The resource management of peer-to-peer networks features many desiderata: time efficiecy, performance, scalability, ease of maintenance, reliability and usability. There are a variety of techniques used to achieve these goals. For example, a solution to evaluate the performance is to make a list of the crucial features surrounding the issue of performance in distributed network systems then to check which feature is the important contributing factor to this issue. Finally, 5 features are identified from this list, and they seem to capture the essence of the proposed enhancements to improve resource management in peer-to-peer networks.

Performance. This is the total time for data read, insert and delete operations. Factors include the locality of data, the efficiency of the locating algorithm, and the efficiency of the routing protocol.

Scalability. This includes the ability of the network system to remain traceable with an increasing number of nodes and data elements. Factors include the balance of space complexity with time complexity.

Maintenance. This includes the amount of manpower unit required to maintain the network system. Factors include the amount of data and topology management that is automated, and the complexity of the code, the data representations, and the network structure.

Reliability. This includes the failure prevention within the network system and the structure of recovery if any failure occurred. Factors include data replication, node failure detection and recovery and finally the existence of multiple guarantees for location information to avoid a single point of failure. Another issue is the availability of multiple paths to data.

Usability. This includes the ease of use, availability of control options, and variety of quality services that the network system offers to the end-user. Factors include the flexibility of the querying of the network system and simplicity of the user interface.

There are five features that can capture the essence of the proposed most important contributing factors to the desirable traits. They are:

Naming. This is the method used to represent shared data objects, network addresses of the nodes, and the structure of routing requests across the network. An appropriate addressing scheme works hand in hand with the algorithms used to increase performance. Hierarchical name spaces increase the network system's scalability in the long-term. A well-structured name space can also be more traceable for a human operator that eases maintenance. Semantic flexibility of naming allows for a variety of query patterns that enhances usability.

Structuring. It includes the organization of the topology and data structures maintained at each node that are used for locating and routing. An efficient structured network system minimizes storage requirements, a key factor in enhancing scalability of the network system.

Locating & Routing. These are the algorithms used to locate data and route to a server. Efficient algorithms minimize overhead of requests/queries and increase both scalability and performance.

Data Managing. This includes the ability to add, delete, replicate, and dynamically shift the location of data between nodes. This affects performance because it allows the network system to exploit locality and balance the load by distributing data to less congested nodes. It allows the network system to scale by relocating data to maximize storage. It allows for reliability by relocating data in case of node failure. Replication also increases reliability by increasing redundancy and locality.

Topological Updating. This includes the abilities to add links, add or delete nodes in the network. This allows performance to be increased by structuring the network to shorten the distance between clients and data nodes. It allows for the network system to be decentralized and avoid the problems of a centralized server. Automatic restructuring of the topology based upon usage minimizes human effort to perform those tasks, easing maintenance.

2 Napster and Gnutella

Napster [1] and Gnutella [2] are two early Peer-to-Peer networks which use centralized and decentralized servers respectively. These have been some of the most popular peer-to-peer networks. This section will describe the Resource Management in Napster and Gnutella as the basic level networks to compare the approaches and enhancement proposed in the subsequent networks then discussed.

2.1 Napster

Napster is a simple structured centralized network system. With respect to the features given above, Napster offers no enhancements on the basic functionality. With regard to the desired traits it has many serious limitations in all of them. But it was very successful socially. It is a sort of simplest model to contrast the other Peer-to-Peer network systems. It uses a centralized server to create its own flat namespace of host addresses. When a client makes a request to a server, it searches first over the client's assigned server and then begins to search other servers until it finds the correct number of responses e.g. one hundred matching music files. These files are organized according to an array of search criteria.

There are problems with using a centralized server including the fact that there is a single point of failure. Napster does not replicate data. It uses a "keep alive" method to make sure that its directories are accessible. Maintaining a unified view is computationally expensive in a network system like Napster. Scaling up can be a problem. It has been a very socially successful network system though. The focus on Napster as a music sharing network system in which users must be active in order to participate has made it exceedingly popular. Regarding routing, it is simply a centralized directory system using Napster servers.

2.2 Gnutella

Gnutella is one of the earliest peer-to-peer file sharing network systems that are completely decentralized. In Gnutella, each node is identified by its IP address and connected to some other nodes. All communication is done over the TCP/IP protocol. To join to the network, the new node needs to know the IP address of one node that is already in the network system. It first broadcasts a "join" message via that node to the whole network system. Each of these nodes then responds to indicate its IP address, how many files it is sharing, and how much space those files take up. So, in connecting, the new node immediately knows how much is available on the network to search through. Gnutella uses file name as the key. Once a search message is sent out to request a name match, it is propagated through the network. Each node that has matching terms passes back its result set. Each node handles the search query in its own way. To save on bandwidth, a node does not have to respond to a query if it has no matching items. The node also has the option of returning only a limited result set. After the client node receives responses from other nodes, it uses HTTP to download the files it wants. Gnutella is completely decentralized. So there's no single point of failure and the scalability is also a little better than Napster. But the nodes are organized loosely, so the costs for node joining and searching are O(N), which means that Gnutella cannot grow to a very large scale.

3 Chord

Chord [5] is a distributed lookup protocol designed by MIT. It supports fast data locating and node joining/leaving. The Chord protocol supports just one operation: given a key, it maps the key onto a node. Depending on the application, that node might be responsible for storing a value associated with the key. Chord uses a variant of consistent hashing [6], [7] to assign keys to Chord nodes. Consistent hashing tends to balance load, since each node receives roughly the same number of keys, and involves relatively little movement of keys when nodes join and leave the network system. Previous work on consistent hashing assumed that nodes were aware of most other nodes in the network system, making it impractical to scale to large number of nodes. In contrast, each Chord node needs "routing" information about only a few other nodes. Because the routing table is distributed, a node resolves the hash function by communicating with a few other nodes. In the steady state, in an N-node network system, each node maintains information only about O(logN) other nodes, and resolves all lookups via O(logN) messages to other nodes. Chord maintains its routing information as nodes join and leave the network system; with high probability each such event results in no more than O($\log^2$N) messages. Three features that distinguish Chord from many other Peer-to-Peer lookup protocols are its simplicity, provable correctness, and provable performance. Chord is simple, routing a key through a sequence of O(logN) other nodes toward the destination. A Chord node requires information about O(logN) other nodes for efficient routing, but performance degrades gracefully when that information is out of date. This is important in practice because nodes will join and leave arbitrarily, and consistency of even O(logN) state may be hard to maintain. Only one piece of information per node need be correct in order for Chord to guarantee correct (though slow) routing of queries; Chord has a simple algorithm for maintaining this information in a dynamic environment.

The Chord protocol specifies how to find the locations of keys, how new nodes join the network system, and how to recover from the failure (or planned departure) of existing nodes. This section describes a simplified version of the protocol that does not handle concurrent joins or failures.

3.1 Overview of Chord

At its heart, Chord provides fast-distributed computation of a hash function mapping keys to nodes responsible for them. It uses consistent hashing, which has several good properties. With high probability the hash function balances load (all nodes receive roughly the same number of keys). Also with high probability, when an Nth node joins (or leaves) the network, only an O(1/N) fraction of the keys are moved to a different location— this is clearly the minimum necessary to maintain a balanced load.

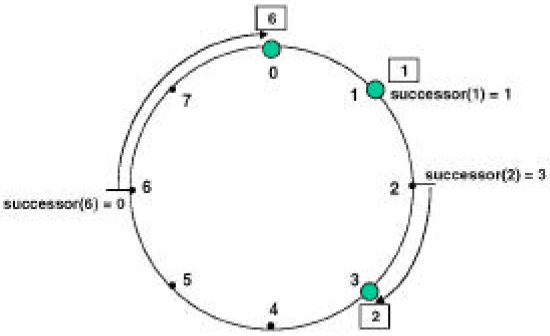


Figure 1: An identifier circle consisting of the three nodes 0, 1,and 3. In this example, key 1 is located at node 1, key 2 at node3, and key 6 at node 0.

Chord improves the scalability of consistent hashing by avoiding the requirement that every node know about every other node. A Chord node needs only a small amount of "routing" information about other nodes. Because this information is distributed, a node resolves the hash function by communicating with a few other nodes. In an N-node network, each node maintains information only about O(logN) other nodes, and a lookup requires O(logN) messages. Chord must update the routing information when a node joins or leaves the network; a join or leave requires O($\log^2$N) messages.

3.2 Consistent Hashing

The consistent hash function assigns each node keys and an m-bit *identifier* using a base hash function such as SHA-1 [8]. A node's identifier is chosen with hashing the node's IP address. Hashing the key produces a key identifier, as well. The term "key" is used to refer to both the original key and its image under the hash function, as the meaning will be clear from context. Similarly, the term "node" will refer to both the node and its identifier under the hash function. The identifier length m must be large enough to make the probability of two nodes or keys hashing to the same identifier negligible.

Consistent hashing assigns keys to nodes as follows. Identifiers are ordered in an *identifier circle* modulo 2^m. Key k is assigned to the first node whose identifier is equal to or follows (the identifier of) *k* in the identifier space. This node is called the successor node of key k, denoted by *successor*(*k*). If identifiers are represented as a circle of numbers from *0* to *2^{m}-1* then *successor*(*k*) is the first node clockwise from *k*.

Figure 2 shows an identifier circle with m = 3. The circle has three nodes: 0, 1, and 3. The successor of identifier 1 is node 1, so key 1 would be located at node 1. Similarly, key 2 would be located at node 3, and key 6 at node 0.

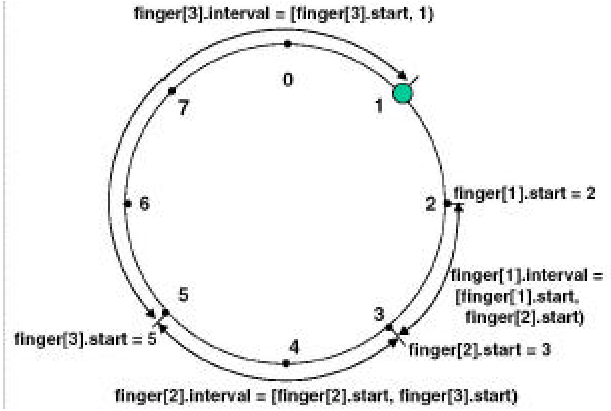

finger[3].interval = [finger[3].start, 1)
0
1
7
6
2
finger[1].start = 2
finger[1].interval = [finger[1].start, finger[2].start)
5
3
finger[3].start = 5
4
finger[2].start = 3
finger[2].interval = [finger[2].start, finger[3].start)

Figure 2 The finger intervals associated with node 1.

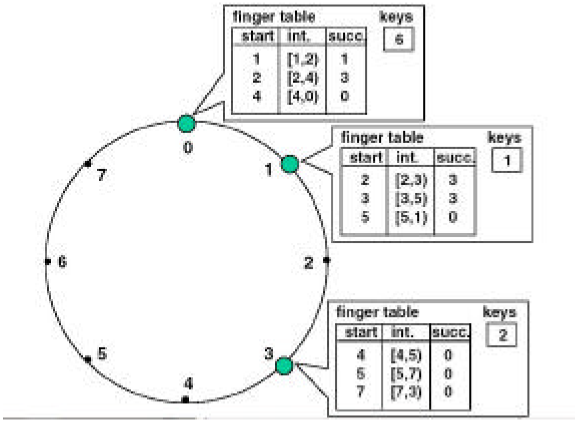


Figure 3 Finger tables and key locations for a net with nodes 0, 1, and 3, and keys 1, 2, and 6.

Consistent hashing is designed to let nodes enter and leave the network with minimal disruption. To maintain the consistent hashing mapping when a node *n* joins the network, certain keys previously assigned to *n's successor* now become assigned to *n*. When node *n* leaves the network, all of its assigned keys are reassigned to *n's successor*. No other changes in assignment of keys to nodes need occur. In the example above, if a node were to join with identifier 7, it would capture the key with identifier 6 from the node with identifier 0.

THEOREM 1. For any set of N nodes and K keys, with high probability:

1. Each node is responsible for at most (1 + e) K/N keys
2. When an (N+1)st node joins or leaves the network, responsibility for O(K/N) keys changes hands (and only to or from the joining or leaving node).

When consistent hashing is implemented as described above, the theorem proves a bound of e = O(logN). The consistent hashing paper shows that e can be reduced to an arbitrarily small constant by having each node run O(logN) "virtual nodes" each with its own identifier.

The phrase "with high probability" bears some discussion. A simple interpretation is that the nodes and keys are randomly chosen, which is plausible in a non-adversarial model of the world. The probability distribution is then over random choices of keys and nodes, and says that such a random choice is unlikely to produce an unbalanced distribution. One might worry, however, about an adversary who intentionally chooses keys to all hash to the same identifier, destroying the load balancing property. The consistent hashing paper uses "k-universal hash functions" to provide certain guarantees even in the case of nonrandom keys.

3.3 Scalable Key Location

A very small amount of routing information suffices to implement consistent hashing in a distributed environment. Each node need only be aware of its successor node on the circle. Queries for a given identifier can be passed around the circle via these successor pointers until they first encounter a node that succeeds the identifier; this is the node the query maps to. A portion of the Chord protocol maintains these successor pointers, thus ensuring that all lookups are resolved correctly. However, this resolution scheme is inefficient: it may require traversing all N nodes to find the appropriate mapping. To accelerate this process, Chord maintains additional routing information. This additional information is not essential for correctness, which is achieved as long as the successor information is maintained correctly.

As before, let *m* be the number of bits in the key/node identifiers. Each node, *n*, maintains a routing table with (at most) *m* entries, called the finger table. The *i*th entry in the table at node *n* contains the identity of the first node, *s*, that succeeds *n* by at least 2^{i}-1on the identifier circle, i.e., *s* = *successor* (n + 2^{i-1}), where 1 = *i* = *m* (and all arithmetic is modulo 2^{m}). Node *s* is called the *i*th finger of node *n*, and we denote it by *n.finger[i].node* (see **Table 1**). A finger table entry includes both the Chord identifier and the IP address (and port number) of the relevant node. Note that the first finger of *n* is its immediate successor on the circle, for convenience it is referred to as the successor rather than the first finger.

Notation	Definition
finger[k].start	*$(n + 2^{k-1})$mod 2^m, 1=k=m*
.interval	*(finger[k].start, finger[k+1].start)*
.node	First node = n.finger[k].start
Successor	The next node on the identifier circle; *finger[1].node*
Predecessor	The previous node on the identifier circle

Table 1 Definition of variables for node *n*, using *m*-bit identifiers

In the example shown in **Figure 4**, the finger table of node 1 to the successor nodes of identifiers $(1 + 2^0)$ mod $2^3 = 2$, $(1 + 2^1)$ mod $2^3 = 3$, and $(1 + 2^2)$ mod $2^3 = 5$, respectively. The successor of identifier 2 is node 3, as this is the first node that follows 2, the successor of identifier 3 is (trivially) node 3, and the successor of 5 is node 0. This scheme has two important characteristics.

1. Each node stores information about only a small number of other nodes, and knows more about nodes closely following it on the identifier circle than about nodes farther away.
2. A node's finger table generally does not contain enough information to determine the successor of an arbitrary key k. For example, node 3 in Figure 3 does not know the successor of 1, as 1's successor (node 1) does not appear in node 3's finger table.

What happens when a node n does not know the successor of a key *k*? If *n* can find a node whose ID is closer than its own to *k*, that node will know more about the identifier circle in the region of k than *n* does. Thus *n* searches its finger table for the node *j* whose ID most immediately precedes *k*, and asks *j* for the node it knows whose ID is closest to *k*. By repeating this process, *n* learns about nodes with IDs closer and closer to *k*.

The pseudo code that implements the search process is shown in **Figure 5**. The notation *n.foo*() stands for the function *foo*() being invoked at and executed on node *n*. Remote calls and variable references are preceded by the remote node identifier, while local variable references and procedure calls omit the local node. Thus *n.foo*() denotes a remote procedure call on node *n*, while *n.bar*, without parentheses, is an RPC to lookup a variable bar on node *n*.

find successor works by finding the immediate predecessor node of the desired identifier; the successor of that node must be the successor of the identifier. Implement *find_predecessor* explicitly, because it is used later to implement the join operation.

When node *n* executes *find_predecessor*, it contacts a series of nodes moving forward around the Chord circle towards id. If node *n* contacts a node *n'* such that id falls between *n*' and the successor of *n'*, find predecessor is done and returns *n'*. Otherwise node *n* asks *n*' for the node *n*' knows about that most closely preceeding id. Thus the algorithm always makes progress towards the predecessor of id.

As an example, consider the Chord ring in **Figure 4**. Suppose node 3 wants to find the successor of identifier 1. Since 1 belongs to the circular interval (7; 3), it belongs to 3:*finger*[3]:*interval*; node 3 therefore checks the third entry in its finger table, which is 0. Because 0 preceeds 1, node 3 will ask node 0 to find the successor of 1. In turn, node 0 will infer from its finger table that 1's successor is the node 1 itself, and return node 1 to node 3.

The finger pointers at repeatedly doubling distances around the circle cause each iteration of the loop in *find_predecessor* to halve the distance to the target identifier. From this intuition follows a theorem:

THEOREM 2. With high probability (or under standard hardness assumptions), the number of nodes that must be contacted to find a successor in an N-node network is O(logN).

PROOF. Suppose that node *n* wishes to resolve a query for the successor of *k*. Let *p* be the node that immediately preceeds *k*.

If *n?p*, *n* forwards its query to the closest predecessor of *k* in its finger table. Suppose that node *p* is in the *i*th finger interval of node *n*. Then since this interval is not empty, node *n* will finger some node *f* in this interval. The distance (number of identifiers) between *n* and *f* is at least 2^i-1. But *f* and *p* are both in *n*'s *i*th finger interval, which means the distances between them is at most 2^i-1. So, it means *f* is closer to *p* than to *n*, or equivalently, that the distance from *f* to *p* is at most half the distance from *n* to *p*. If the distance between the node handling the query and the predecessor *p* halves in each step, and is at most 2^m initially, then within *m* steps the distance will be one.

In fact, as discussed above, it's assumed that node and key identifiers are random. In this case, the number of forwarding necessary will be O(logN) with high probability. After logN forwarding, the distance between the current query node and the key *k* will be reduced to at most 2^m/N. The expected number of node identifiers landing in a range of this size is 1, and it is O(logN) with high probability. Thus, even if the remaining steps advance by only one node at a time, they will cross the entire remaining interval and reach key *k* within another O(logN) steps.

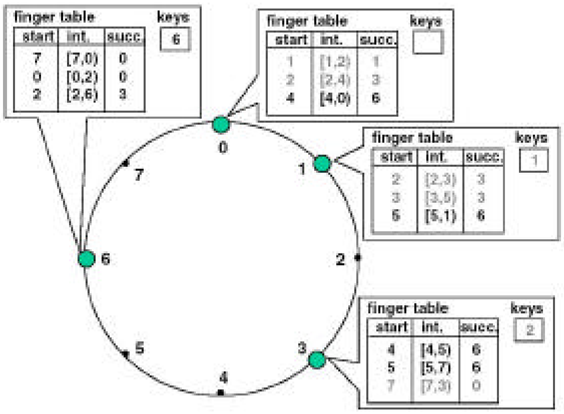


Figure 4 Finger tables and key locations after node 6 joins.

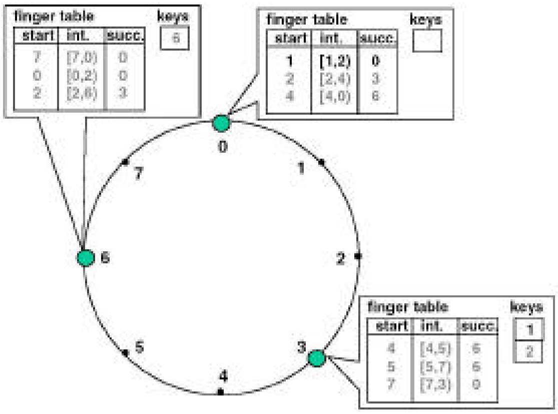


Figure 5 Finger tables and key locations after node 3 leaves. Changed entries are shown in black, and unchanged in gray.

3.4 Node Joins

In a dynamic network, nodes can join (and leave) at any time. The main challenge in implementing these operations is preserving the ability to locate every key in the network. To achieve this goal, Chord needs to preserve two invariants:

1. Each node's successor is correctly maintained.
2. For every key *k*, node *successor(k)* is responsible for *k*.

In order for lookups to be fast, it is also desirable for the finger tables to be correct.

THEOREM 3. With high probability, any node joining or leaving an N-node Chord network will use O(log2 N) messages to re-establish the Chord routing invariants and finger tables.

To simplify the join and leave mechanisms, each node in Chord maintains a *predecessor pointer*. A node's predecessor pointer contains the Chord identifier and IP address of the immediate predecessor of that node, and can be used to walk counterclockwise around the identifier circle.

To preserve the invariants stated above, Chord must perform three tasks when a node *n* joins the network:

1. Initialize the predecessor and fingers of node *n*.
2. Update the fingers and predecessors of existing nodes to reflect the addition of *n*.
3. Notify the higher layer software so that it can transfer state (e.g. values) associated with keys that node *n* is now responsible for.

It was assumed that the new node learns the identity of an existing Chord node *n'* by some external mechanism. Node *n* uses *n'* to initialize its state and add itself to the existing Chord network, as follows.

Initializing fingers and predecessor: Node *n* learns its predecessor and fingers by asking *n'* to look them up. Naively performing finds successor for each of the *m* finger entries would give a runtime of O(mlogN). To reduce this, *n* checks whether the *i*th finger is also the correct (*i* + *1*)th finger, for each *i*. This happens when *finger[i]:interval* does not contain any node, and thus *finger[i]:node* = *finger[i+1]:start*. It can be shown that the change reduces the expected (and high probability) number of finger entries that must be looked up to O(logN), which reduces the overall time to O($\log^2$N). As a practical optimization, a newly joined node *n* can ask an immediate neighbor for a copy of its complete finger table and its predecessor. *n* can use the contents of these tables as hints to help it find the correct values for its own tables, since *n*'s tables will be similar to its neighbors'. This can be shown to reduce the time to fill the finger table to O(logN).

Updating fingers of existing nodes: Node *n* will need to be entered into the finger tables of some existing nodes. For example, in **Figure 5**, node 6 becomes the third finger of nodes 0 and 1, and the first and the second finger of node 3.

The pseudo code of the *update_finger_table* function updates the existing finger tables. For example, node *n* will become the *i*th finger of node *p* if and only if *p* preceeds *n* by at least 2^{i-1}, and the *i*th finger of node *p* succeeds *n*. The first node, *p*, which can meet these two conditions, is the immediate predecessor of *n*-2^{i-1}. Thus, for a given *n*, the algorithm starts with the *i*th finger of node *n*, and then continues to walk in the counter-clockwise direction on the identifier circle until it encounters a node whose *i*th finger precedes *n*.

When a node joins the network, the number of nodes that need to be updated is O(log N) with high probability. Finding and updating these nodes takes O($\log^2$N) time. A more sophisticated scheme can reduce this time to

O(logN); however, it will bring a much more complex implementation of the algorithm.

Transferring keys: The last operation that has to be performed when a node *n* joins the network is to move responsibility for all the keys for which node *n* is now the successor. Exactly what this entails depends on the higher-layer software, but typically it would involve moving the data associated with each key to the new node. Node *n* can become the successor only for keys that were previously the responsibility of the node immediately following *n*, so *n* only needs to contact that one node to transfer responsibility for all relevant keys.

3.5 Evaluation

Naming each machine is assigned an m-bit nodeId, which is generated by hashing its IP address. Each data record (*k*, *v*) has its unique key *k*. In Chord, it is also assigned an m-bit ID by hashing the key, *p=hash(k).* This ID is used to indicate the location of the data.

Structuring the entire possible $N=2^m$ nodeIds are ordered in a one-dimensional circle; the machines are mapped to this virtual circle according to their nodeIds. For each nodeId, the first physical machine on its clockwise side is called its successor node, or *successor(nodeId).* Each data record (*k*, *v*) has an identifier *p=hash(k),* which indicates the virtual position in the circle. The data record (*k, v)* is stored in the first physical machine clockwise from *p*. This machine is called the successor node of P, or *successor(p*). To do routing efficiently, each machine contains part of the mapping information. In the view of each physical machine, the virtual cycle is partitioned into 1+logN segments: itself, and logN segments with length 1, 2, 4, …, N/2. The machine maintains a table with logN entries; each entry contains the information for one segment: the boundaries and the successor of its first virtual node. In this way, each machine only need O(logN) memory to maintain the topology information. And it appears that the information is sufficient for fast locating/routing.

Locating & Routing On query for a record with key *k*, the virtual position is first calculated: *p=hash(k).* The locating can start from any physical machine. Using the mapping table, the successor of the segment that contains P is selected to be the next router until P lies between the start of the segment and the successor (this means the successor is also P's successor, i.e., the target). The distance between the target and the current machine will decrease by half after each hop. Thus the routing time is O(logN).

Data Managing For high availability, the data can be replicated using multiple hash functions; it's also possible to replicate the data at the *r* machines succeeding its data ID. All the data operation is in O(logN) time.

Topology Updating In Chord, machines can join and leave at any time. For normal node arrival and departure, the cost is $O(\log^2 N)$(with complex algorithm it could be O(logN)) with high probability, but in the worst case, the cost is O(N). A node failure can also be detected and recovered automatically if each node maintains a "successor-list" of its r nearest successors on the Chord ring.

4 Pastry

Pastry [3] is a generic peer-to-peer content location and routing network system based on a self-organizing overlay network of nodes connected via the Internet. It is completely decentralized, scalable, fault-resilient, and reliably routes a message to the live node with a nodeId numerically closest to a key with that message; it automatically adapts to the arrival, departure and failure of nodes.

Any Internet-connected host that runs the Pastry software and has proper credentials can participate in the overlay network.

Each Pastry node has a unique, 128-bit nodeId. The set of existing nodeIds is uniformly distributed; this can be achieved, for instance, by basing the nodeId on a secure hash of the node's public key or IP address. Given a message and a key, Pastry reliably routes the message to the Pastry node with the nodeId that is numerically closest to the key, among all live Pastry nodes.

Assuming a Pastry network consisting of N nodes, Pastry can route to any node in less than $[\log_{2^b} N]$ steps on average (*b* is a configuration parameter with typical value 4). With concurrent node failures, eventual delivery is guaranteed unless *l* /2 or more nodes with adjacent nodeIds fail simultaneously (*l* is an even integer parameter with a typical value of 16).

The tables required in each Pastry node have only $(2^b-1)*(\log_2{}^b N)+l$ entries, where each entry maps a nodeId to the associated node's IP address. Moreover, after a node failure or the arrival of a new node, the routing tables can be restored with the cost of needing to exchange $O(\log_2{}^b N)$ messages.

For the purposes of routing, nodeIds and keys are thought of as a sequence of digits with base 2^b. A node's routing table is organized into $[\log_2{}^b N]$ rows with 2^b-1entries each. The 2^b-1 entries in row *n* of the routing table each refer to a node whose nodeId matches the present node's nodeId in the first *n* digits, but whose *n*+1th digit has one of the 2^b-1possible values other than the *n*+1th digit in the present node's id. The uniform distribution of nodeIds ensures an even population of the nodeId space; thus, only $[\log_2{}^b N]$ levels are populated in the routing table. Each entry in the routing table refers to one of potentially many nodes whose nodeId have the appropriate prefix. Among such nodes, the one closest to the present node (according to a scalar proximity metric, such as the round trip time) is chosen.

In addition to the routing table, each node maintains IP addresses for the nodes in its leaf set, i.e., the set of nodes with the *l* /2 numerically closest larger nodeIds, and the *l* /2 nodes with numerically closest smaller nodeIds, relative to the present node's nodeId.

Figure 2 shows the path of an example message. In each routing step, the current node normally forwards the message to a node whose nodeId shares with the key a prefix that is at least one digit (or b bits) longer than the prefix that the key shares with the current nodeId. If no such node is found in the routing table, the message is forwarded to a node whose nodeId shares a prefix with the key as long as the current node, but is numerically closer to the key than the current nodeId. Such a node must exist in the leaf set unless the nodeId of the current node or its immediate neighbor is numerically closest to the key, or *l* /2 adjacent nodes in the leaf set have failed concurrently.

4.1 Locality

Locality properties of Pastry have many interesting features, i.e., the properties of Pastry's routes with respect to the proximity metric. The proximity metric is a scalar value that reflects the "distance" between any pair of nodes, such as the round trip time. It is assumed that a function exists that allows each Pastry node to determine the "distance" between it and a node with a given IP address.

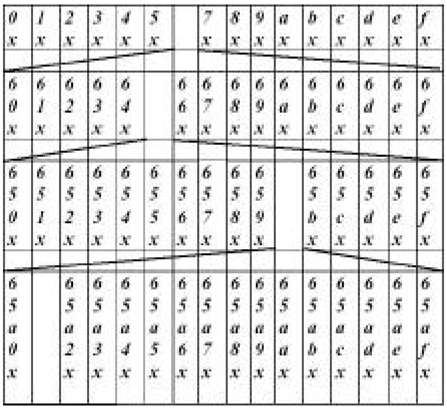
Two of Pastry's locality properties, which are relevant to Scribe, are the advantage of Pastry. The *short routes* property concerns the total distance, in terms of the proximity metric, that messages travel along Pastry routes. Recall that each entry in the node routing tables is chosen to refer to the nearest node, according to the proximity metric, with the appropriate nodeId prefix. As a result, in each step a message is routed to the nearest node with a longer prefix match. (Simulations [4] performed on several network topology models show that the average distance traveled by a message is between 1.59 and 2.2 times the distance between the source and destination in the underlying Internet).

The *route convergence* property is concerned with the distance traveled by two messages sent to the same key before their routes converge. Simulations show that, given our network topology model, the average distance traveled by each of the two messages before their routes converge is approximately equal to the distance between their respective source nodes.

4.2 Node addition and failure

A key design issue in Pastry is how to efficiently and dynamically maintain the node state, i.e., the routing table, leaf set and neighborhood sets, in the presence of node failures, node recoveries, and new node arrivals. Briefly, an arriving node with the newly chosen nodeId X can initialize its state by contacting a nearby node A (according to the proximity metric) and asking A to route a special message using X as the key. This message is routed to the existing node Z with nodeId numerically closest to X1. X then obtains the leaf set from Z, and the *i*th row of the routing table from the *i*th node encountered along the route from A to Z. One can show that using this information, X can correctly initialize its state and notify nodes that need to know of its arrival.

To handle node failures, neighboring nodes in the nodeId space (which are aware of each other by virtue of being in each other's leaf set) periodically exchange 'keep alive' messages. If a node is unresponsive for a period T, it is presumed failed. All members of the failed node's leaf set are then notified and they update their leaf sets. Since the leaf sets of nodes with adjacent nodeIds overlap, this update is trivial. A recovering node contacts the nodes in its last known leaf set, obtains their current leaf sets, updates its own leaf set and then notifies the members of its new leaf set of its presence. Routing table entries that refer to failed nodes are repaired lazily.



0x	1x	2x	3x	4x	5x		7x	8x	9x	ax	bx	cx	dx	ex	fx
60x	61x	62x	63x	64x		66x	67x	68x	69x	6ax	6bx	6cx	6dx	6ex	6fx
650x	651x	652x	653x	654x	655x	656x	657x	658x	659x		65bx	65cx	65dx	65ex	65fx
65a0x		65a2x	65a3x	65a4x	65a5x	65a6x	65a7x	65a8x	65a9x	65aax	65abx	65acx	65adx	65aex	65afx

Figure 6 Routing table of a Pastry node with nodeId 65a1x, b = 4. Digits are in base 16. x represents an arbitrary suffix. The IP address associated with each entry is not shown.

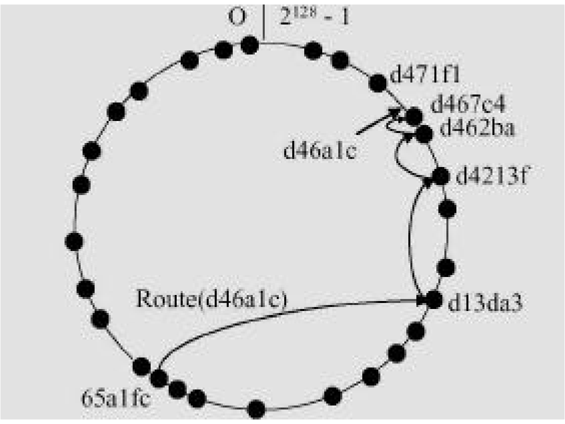


Figure 7 Routing a message from node 65a1fc with key d46a1c. The dots depict live nodes in Pastry's circular namespace.

4.3 Evaluation

Naming Each node in the Pastry peer-to-peer overlay network has a unique 128-bit nodeId, this nodeId is assigned randomly when a node joins the network system by computing a cryptographic hash of the node's public key or its IP address. With this naming mechanism, Pastry makes an important assumption that nodeIds are generated such that the resulting set of nodeIds is uniformly distributed in the nodeId space. Each data also has a 128-bit key. This key can be the original key, or generated by a hash function. The data is stored in the node whose id is numerically closest to it key.

Structuring Each Pastry node maintains 3 sets of information: a routing table, a neighborhood set and a leaf set.

- **Routing table** Assuming a network consisting of N nodes, a node's routing table is organized into logN rows with 2^b-1 entries on each row. Every *i*th row shares first *i* digits with present nodeId, but *i*+1th digit has any one of 2^b-1 possible values. If there are more than 2^b-1 qualified nodes, the closest 2^b-1 nodes will be selected, according to proximity metric. The routing table is used for incremental routing.
- **Neighborhood set** Neighborhood set contains nodeIds and IP addresses of nodes that are closest (physically and with proximity) to the local node.
- **Leaf set** The leaf sets has *l* /2 numerically closest larger and *l* /2 closest smaller nodes and is used for direct routing.

Locating & Routing Given a message, the node first checks to see if the key falls within the range of nodeIds covered by its leaf set. If so, the message is forwarded directly to the destination node, namely the node in the leaf set whose nodeId is closest to the key. If the key is not covered by the leaf set, then the routing table is used and the message is forwarded to a node that shares a common prefix with the key by at least one more digit. In certain cases, it is possible that the appropriate entry in the routing table is empty or the associated node is not reachable, in which case the message is forwarded to a node that shares a prefix with the key at least as long as the present node, and is numerically closer to the key than the present node's nodeId. Such a node must be in the leaf set unless the message has already arrived at the node with numerically closest nodeId.

Data Managing Pastry supports dynamic data object insertion and deletion, but does not explicitly support for mobile objects.

Topology Updating Pastry supports dynamic node join and departure.

5 Comparative Property Analysis

This Section, the detailed information about the features of each network system is given in **Table 2**. The comparisons of these 4 networks' performance, scalability, scalability, reliability, and maintenance are listed.

	Napster	Gnutella	Chord	Pastry
Decentralized	No	Yes	Yes	Yes
Space cost	O(N)	Depends	O(logN)	O(logN)
Data read	O(1)	O(N)	O(logN)	O(logN)
Data insert	O(1)	O(1)	O(logN)	O(logN)
Data delete	O(1)	O(1)	O(logN)	O(logN)
Node insert	O(1)	O(N)	O(logN)	O(logN)
Node delete	O(1)	O(1)	O(logN)	O(logN)
Node failure	–	–	O(logN)	O(logN)
Locality	Yes	–	No	Yes

Table 2 The comparison of each network system.

5.1 Performance

It's clear that Napster is bad because it uses a central server that is likely to be over-loaded. The server needs large storage to maintain the information about all the nodes and data; the response time will increase when the number of nodes and requests exceed the capability of the server. Though Gnutella is completely decentralized, its performance is not satisfying. Because the nodes are organized loosely, the costs for node joining and searching are O(N). All the other network systems perform well. They all have logN-like performance.

5.2 Scalability

Napster needs O(N) storage and computing power, Gnutella needs O(N) routing time cost, so Napster and Gnutella are not satisfying.

5.3 Reliability

Napster has the single point of failure. The centralized server also is a easy target for a DoS attack. All the other network systems are better than Napster by using decentralized organization to eliminate the single point of failure. The additional mechanisms to achieve reliability are listed below:

Pastry: Routing in Pastry can be random, i.e., the choice among multiple nodes can be made randomly. In the event of a malicious or failed node along the path, the query may be repeated several times by the client, until a route is chosen that avoids the bad node.

Chord: A good reliability is achieved by maintaining multiple data replicas and multiple successors. In the case of using r=O(logN) successors, even if every node fails with probability 1/2, with high probability the location algorithm can still find the closest living successor to the query key in expected time O(logN).

5.4 Maintenance

For Napster and Gnutella, the nodes are organized loosely, so no work is necessary to making the network system consistent. Pastry, Chord, Tapestry and CAN all support dynamic node arrival and departure.

5.5 Usability

Napster supports the searching for music files. The central server can search its database and find a number of "optimal" files for the user. On the client's view, Gnutella provides a similar function as Napster, but it supports different file types. On the server's view, Gnutella allows each node to decide its own sharing mechanisms on different files. For Pastry and Chord, basically they only support looking up for a data given a key.

6 The polling from end-users

The feeling of the end-users is the most important evaluation criteria, even more important than the result of the simulators. Overnet and eDonkey are the idea examples for the polling of end users' feeling, because almost all of their protocols are the same, except the resource management protocol. And they have almost the same subscribers number. Overnet is base on DHT and Chord-like Ring (Overnet has it's private protocol, but is similar to Chord). eDonkey is a centralized Peer-to-Peer Network system. The polling [9] result is that 64% of the users prefer Overnet. In another words it could be said that end-users like more the Chord-ring

7 References

[1] Napster. http://www.napster.com/

[2] Knowbuddy's Gnutella FAQ, http://www.rixsoft.com/Knowbuddy/gnutellafaq.html

[3] A. Rowstron and P. Druschel, Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. http://research.microsoft.com/~antr/PAST/

[4] M. Castro, P. Druschel, Y. C. Hu, and A. Rowstron. Topology-aware routing in structured peer-to-peer overlay networks, 2002. Submitted for publication.

[5] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. http://www.acm.org/sigcomm/sigcomm2001/p12-stoica.pdf

[6] KARGER, D., LEHMAN, E., LEIGHTON, F., LEVINE, M., LEWIN,D., AND PANIGRAHY, R. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web. In Proceedings of the 29th Annual ACM Symposium on Theory of Computing (El Paso, TX, May 1997), pp. 654–663.

[7] LEWIN, D. Consistent hashing and random trees: Algorithms for caching in distributed networks. Master's thesis, Department of EECS, MIT, 1998. Available at the MIT Library, http://thesis.mit.edu/

[8] FIPS 180-1. Secure Hash Standard. U.S. Department of Commerce/NIST, National Technical Information Service, Springfield, VA, Apr. 1995.

[9] Offical eDonkey forum http://forum.overnet.com/

Annexes

Peer-to-Peer networks/protocols and applications

1. Napster network
   - OpenNap
   - WinMx
   - Napigator
   - FileNavigator
2. Gnutella network
   - Acquisition (Mac OS)
   - BearShare
   - Gnucleus
   - Limewire
   - Morpheus
   - Phex
   - Swapper
   - Shareaza
   - XoloX
   - Gtk-gnutella
3. Chord network
   - CFS (Cooperative File System)
   - i3
4. Pastry network
   - Scribe
   - PAST
   - SQUIRREL
   - SplitStream
   - POST
   - Scrivener

Performance measurements and availability problems in P2P systems

Johanna Antila
Researcher
Otakaari 5 A 02150 ESPOO, Finland
jmantti3@netlab.hut.fi

Abstract

The use of peer-to-peer applications has grown dramatically over the past few years making peer-to-peer systems an increasingly important research topic. Some of the most relevant research problems in peer-to-peer systems are the performance and availability issues. In this paper we focus on performance and availability problems in file sharing peer-to-peer systems. We discuss what parameters should be used in the performance evaluation, how these parameters could be measured and present the most important findings that have been obtained from earlier performance measurements studies. Finally, we discuss how these results may affect future developments of peer-to-peer systems. We propose that new peer selection, replication and caching algorithms should be deviced for improving the performance of current peer-to-peer systems. We also suggest that measurements should be utilized for generating accurate enough models of peer-to-peer systems and that those models could be used to evaluate new algorithms and protocols by simulations.

1 Introduction

During the recent years peer-to-peer applications such as Napster[12], Gnutella[10], Kazaa[11] and Freenet[9] have gained enourmous popularity. This is mainly due to the fact that these applications provide an easy way to download audio, video and image files and different kinds of software from all over the world, either free of charge or with minor costs. The surprisingly fast emergence of peer-to-peer applications has changed the traffic patterns in the Internet dramatically and thus affects how traffic should be handled in the network. It is obvious that considerable amount of research has to be done in order to better understand how these systems behave, what the performance of these systems is at the moment and how the performance could be improved in the future.

In this paper we focus on performance measurements and availability problems in file sharing peer-to-peer systems. The paper is organized as follows: Section 2 describes the basic principles of peer-to-peer architectures. In sections 3 and 4 we present the key parameters that should be used in the performance evaluation of peer-to-peer systems and show different ways to measure them. In section 5 we summarize the most important measurement results that have been obtained up to the present. Finally, in section 6 we discuss what implications these results may have on future developments of peer-to-peer systems. We suggest that some kind of peer selection, caching and replication mechanisms should be used to improve the performance of both content location and retrieval. We also propose that the behavior of peer-to-peer systems should be modeled with the help of extensive measurements in order to enable performance evaluation of new peer-to-peer algorithms and protocols by simulations. Section 7 concludes the paper.

2 Peer-to-peer systems

2.1 Peer-to-peer systems in general

Contrary to the traditional client-server model of communication, in peer-to-peer communication model there is no centralized infrastructure. Ideally, each participant acts both as a client and a server, consuming and contributing resources. Another key feature in these systems is that membership is not static but rather ad-hoc: peers may join or leave the system at arbitrary time. This dynamic nature of peer-to-peer systems introduces many challenges for efficient content location and retrieval.

The peer-to-peer concept may be applied in many contexts. For instance, processing intensive applications may utilize idle cycles in personal computers in order to perform complex computations required e.g. in medicine, bioinformatics and astronomy in a distributed fashion. However, in this paper we will focus exclusively on file sharing peer-to-peer systems due to the fact that currently the file sharing applications generate the largest fraction of peer-to-peer traffic load.

The following sections present three examples of the most popular file sharing peer-to-peer system architectures: Napster, Gnutella and Kazaa. In all these architectures files can be located and exchanged between the participants over the Internet. The files are stored in individual users' computers and the exchange is performed by a direct connection using a HTTP-style protocol. However, there are differences in how the files are located in these systems.

2.2 Napster

Napster is the first and probably the most well known peer-to-peer system. In Napster, a cluster of central servers is used that maintains an index of all the files that are shared by active peers. Each peer is connected to one of these servers. When a peer wants to exchange a file, it first sends a file location query to this server. The servers in the cluster process the query in cooperation and inform the peer about possible locations of the requested file. Some metadata about the download sites such as reported connection bandwidth is also returned along with the locations in order to assist the peer in deciding where from to download the file.

2.3 Gnutella

In Gnutella, an overlay network is used instead of central servers meaning that point-to-point connections are maintained with a set of neighbors. When a peer wants to locate a file, it floods a query packet to all of its neighbors. When the neighbor receives the query, it checks whether it has the requested file. If so, this peer sends a query response packet to the originating peer. The peer also floods the query further in the overlay regardless of whether it had the requested file or not. The overlay is maintained with the help of *ping* and *pong* messages that enable the peers to discover other nodes as they dynamically enter and leave the system. Figure 1 depicts the difference in Gnutella (original version) and Napster architectures. However, it should be noticed that in the latest version of Gnutella dedicated ultra nodes may also be used, which makes Gnutella more like Kazaa (see section 2.4).

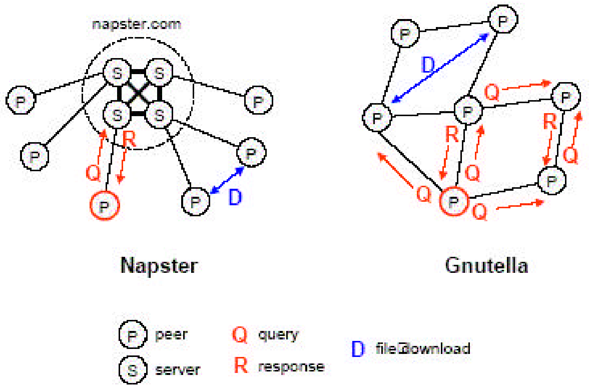


Figure 1 Napster and Gnutella (original version) architectures [1]

2.4 Kazaa

Kazaa is an example of a more recent peer-to-peer system that uses a hybrid architecture, where some peers can be elected as “supernodes”. Supernodes are peers that have fast Internet connections and powerful computers. When a user issues a file query, this query is sent to the nearest supernode. The supernode will then refer the query to other supernodes [11]. The supernodes keep a list of some of the files available at individual Kazaa user’s computers. The actual retrieval of the file occurs by connecting to the host that has the requested file, not through the supernode. In general, operating as a supernode will consume only about 10% of the total CPU power of the peer.

3 Performance metrics of P2P systems

From the user´s point of view the most important performance aspects in a peer-to-peer system are the time required to locate the requested files and the rate at which the located content can be downloaded, i.e. TCP throughput. This means that at network level the following parameters should be measured.

3.1 Latency

Latency is an important parameter in the file location phase, where queries of relatively small size should be processed as quickly as possible. However, it is relevant also in the downloading phase where high volumes of data are transferred since TCP’s congestion control favors flows with short roundtrip times. Also, theoretically TCP throughput in steady-state is inversely proportional to the roundtrip time.

3.2 Available bandwidth

Even though TCP throughput is related to latency, the available bandwidth along the path is even more relevant. Often available bandwidth is approximated by measuring the bottleneck link bandwidth, which defines the “physical” upper limit on the rate at which content can be downloaded. The bottleneck is usually the last-hop link to the peer itself [1]. In reality the available bandwidth depends heavily on the load level of the peer, making bottleneck bandwidth just a coarce approximation.

In many peer-to-peer systems the peers may report their connection bandwidth by themselves. However, in practice all peers do not choose to use this capability. Even if they did, there is an incentive for the peer to misreport its bandwidth in order to discourage other peers from downloading files [1]. Thus measurements should be used to verify the actual bandwidths.

3.3 Availability

Availability can be defined as “the quality of being present or ready for immediate use” [2]. In ad-hoc type peer-to-peer systems where hosts may join and leave systems arbitrarily availability is perhaps the most crucial performance parameter. No matter how low the latency between two hosts would be or how high the bandwidth, files can not be exchanged unless both hosts are currently active. In systems that use personal computers for storing large scientific data sets availability is even more crucial.

In this paper availability is referred to as host availability, which is a continuous function of time rather than a static parameter. In principle, issues such as hardware or software failures also affect availability but the purpose of host availability is mainly to model the activity patterns of the hosts belonging to the peer-to-peer system.

3.4 Query hit ratio

Query hit ratio is an important efficiency metric in the search phase. It is defined as the ratio of successful queries to all queries sent. If query hit ratio is low it means that a lot of bandwidth is wasted for queries that can not be solved. Currently query hit ratios have been estimated mainly by simulations, not by measurements.

4 Measurement methodologies

Performance measurements of existing peer-to-peer systems consist at least of two phases:

1. Finding the peers (or a subset of peers) participating in the system.
2. Conducting the actual measurements by connecting to the peers/subset of peers from a well known host/hosts.

An alternative for finding the peers in an existing peer-to-peer system is to conduct measurements directly between some hosts in the Internet. The advantage of this methodology is that the measurements provide information about the actual performance between the peers, not about performance of the peers with regard to some fixed reference point. However, direct measurements require volunteers to run measurement software on their computers. Furthermore, the measurements do not describe the performance of a "real" peer-to-peer system. Thus the volunteers should be selected carefully so that their characteristics represent the properties of typical participants of a peer-to-peer system.

4.1 Crawlers

Crawlers are pieces of software that aim to discover the peers (or a subset of peers) participating in the system. The basic idea is to generate queries for files and keep a list of peers that are referenced in the responses. In order to discover as many peers as possible, popular file names should be used in the queries. In Gnutella, sending of queries is not necessary since ping/pong messages may be utilized directly for host discovery.
The crawlers may capture either IP-addresses [1] or unique IDs [2]. In [2] the use of unique Ids is proposed instead of IP addresses due to an aliasing phenomenon: According to [2] the ratio of unique host ID to IP address is only 1:4, mainly because of DHCP that may assign a different address for a host when it joins the system at a later time and NAT that uses private IP addresses for hosts behind the NAT boxes. However, the problem in using unique Ids is that only a few peer-to-peer systems support them. One of these systems is the Overnet [19].

4.2 Probers

Most of the measurement studies of peer-to-peer systems have concentrated on characterizing the workload and traffic patterns of peer-to-peer systems ([5], [15], [17], [18]). Only a few extensive papers have been written about the performance measurements of these systems. Up to the present, [1], [2], [3] and [4] are the most prominent research efforts in the field. Even in these papers the performance has been measured quite indirectly. As far as we know, there does not exist any paper where user level performance parameters such as file search time and file download time would have been measured in large scale peer-to-peer systems.

This Section describes how different parameters are measured in [1], [2], [3] and [4]. In [1], a dedicated measurement host is used that measures directly certain performance parameters of the system by connecting to the peers discovered by the crawlers. In [2] and [3], a similar methodology is used with the difference that since these papers concentrate on measuring availability, a subset of peers is probed at regular intervals. In [4], a measurement software called PeerMetric is developed, which includes a client and a server component and enables direct performance measurements between peers. The PeerMetric clients reside on the users computers and support many measurement tests such as pings to arbitrary Internet hosts, application level UDP pings, UDP packet pairs and TCP transfers to/from other peers and HTTP transfers of objects. The server side is responsible for keeping track of the clients that are online and invoking different measurement tests. 25 volunteer peers are used in the measurements.

4.2.1 Measuring latency

In [1], latency is measured by sending a small TCP packet from the measurement host to the peer and measuring the time it takes for the packet to go through the network. Although the delays depend on the location of the measurement host, the authors believe that the distribution of delays would be similar regardless of the measurement point. In [4], latencies are measured directly between certain peers using application level UDP pings (ordinary ICMP pings could be disabled by NATs or by the peers themselves). In [13], another interesting methodology, King, for measuring latencies is introduced that could possibly be applied also in performance evaluation of peer-to-peer systems. In King, the idea is to utilize DNS name servers that are topologically close to the end hosts for which the latency should be measured. King sends a recursive DNS query to one of these servers, asking it to resolve a name that belongs to a domain for which the other server is responsible. Thus the latency between the hosts will be approximated by the latency between the DNS servers meaning that no measurement infrastructure has to be deployed in the end hosts.

4.2.2 Measuring bandwidth

In both [1] and [4] bottleneck link bandwidth is measured between a well-connected server and the peers. The idea is that since the other end is well-connected, the bottleneck will most likely be the last-hop to the peer at the other end. In [4] simple packet-pair test enabled by the PeerMetric were run both in upstream and downstream directions. However, the results may be affected by interfering cross traffic. In [1], a new tool called Sprobe was developed for measuring the bottleneck link bandwidth. Sprobe is also based on packet-pair dispersion technique and is able to measure both upstream and downstream bottleneck link bandwidths. However, contrary to PeerMetric, this tool detects interfering cross-traffic thus improving the accuracy of the measurements. In [4] TCP throughput has also been measured directly.

4.2.3 Measuring availability

In [1] and [3] availability is characterized by measuring the distributions of node uptimes. The nodes are supposed to be in one of the three states:

1. **Offline** – The peer does not respond to TCP SYN packets, because it can not handle more requests at the moment or because it is not connected to the Internet.
2. **Inactive** – The peer responds to TCP SYN packets with RST's, because it is connected to the Internet but not to the peer-to-peer system.
3. **Active** – The peer responds to TCP SYN packets with a SYN/ACK packet and is thus connected to the peer-to-peer system.

Thus, the state of the peer can be discovered by sending a TCP/SYN packet and waiting how it responds. This process should be repeated at regular intervals in order to gain the node uptimes. However, it should be noted that this kind of measurement is possible only if the peer is not behind a NAT or a firewall. In [1], both Internet host uptime and Gnutella/Napster host uptime has been measured. An Internet host is said to be up if it is either in the inactive or in the active state. The Gnutella/Napster host on the other hand is up only if it is in the active state and is able to responds to application-level requests. In [4], only host up times have been measured.

In [2], 2400 hosts in an Overnet system were selected randomly for probing from the hosts that were discovered by the crawler. These hosts were probed at 20 minute interval during 7 days. Contrary to [1] and [3], normal Overnet protocol traffic was used for probing instead of TCP SYN packets. This eliminates the effect of address aliasing due to DHCP and NAT, since unique Ids can be used. Availability was calculated by dividing the number of probes that a host responded to by the total number of probes sent to that host. Different time intervals were used for averaging.

4.3 Role of simulations in performance evaluation

Measurements can be used for evaluating the performance of existing peer-to-peer systems. However, it is extremely difficult to evaluate the performance of new peer-to-peer algorithms and protocols with measurements, especially if testing the algorithms would require each peer in the network to implement the algorithm. Thus simulations should be used to complement measurements as has long been done in other fields of Internet research. However, currently there are no proper models and practices for simulating peer-to-peer systems. This is mainly because peer-to-peer applications are quite a new phenomenon and thus their characteristics are not yet well understood. Measurements would provide a means for gathering relevant information from the system so that models regarding the behavior of the peers and the properties of shared content could be constructed, much in the same way that models have been developed e.g. about the web traffic. Parameters that should be measured for constructing the models include e.g.

- Number of files shared by the peers
- Content categories and popularity of files within the category
- Distribution of node uptimes and session durations
- Query rates and interest levels in content categories
- Topology of the overlay network, bandwidths and latencies of the peers
- Peer selection process

Some of these parameters have been modeled in [16], where the authors have developed a simple query cycle simulator for simulating peer-to-peer networks. Also e.g. in [6], [8] and [15] performance of new peer-to-peer algorithms has been studied with the help of simulations. However, all of these are just initial steps towards more refined practices.

5 Performance measurement results

5.1 Latency

Figure 1 shows the CDF of measured latencies of Gnutella peers according to [1]. It can be observed that there is large heterogeneity in the latencies among the peers. Furthermore, a significant part of the peers will suffer from high latency: for 20 % of the peers the latency is at least 280 ms [1]. The latency depends much on whether the peers are located in the same part of the continent, in opposite parts of the continent or if the peers are trans-oceanic. It should also be noticed that in [1] the measurements were performed with respect to a fixed measurement point and thus the absolute delay values would not stay the same if the location of the measurement point would be changed.

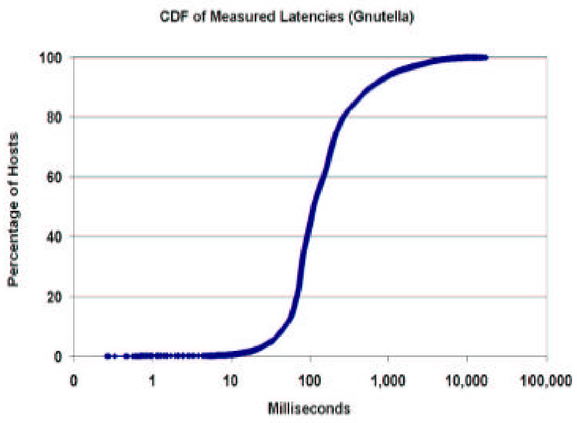


Figure 1 Measured latencies to Gnutella peers [1]

Figure 2 shows the latency results from the measurements of [4]. The absolute values are different compared to Figure 1 since in [4] the measurements were conducted directly between the peers. However, the form of the delay CDF is still similar, revealing large heterogeneity between the peers. Some difference can also be observed between the delays of DSL and cable modem hosts: the delay for cable modem hosts is larger and more variable, probably due to cable medium’s shared nature [4].

5.2 Bandwidth

Upstream bottleneck bandwidth determines the upper limit on the rate at which the peer can serve content. In Figure 3 the CDF of both upstream and downstream bottleneck bandwidths for Gnutella peers are shown based on the measurements of [1]. It can be seen that approximately 20 % of the peers have upstream bottleneck bandwidth of less than 100 kbps. Clearly, these peers would not be suitable for serving many file requests. The downstream bottleneck bandwidths on the other hand tend to be larger, as can be observed from Figure 3. This is due to the fact that many peers use asymmetric access technology, such as ADSL or cable modem. Asymmetric access technology is justified if a peer mainly downloads content, but in peer-to-peer networks symmetric access technology would be a better alternative since a peer should be able to act both as a client and a server.

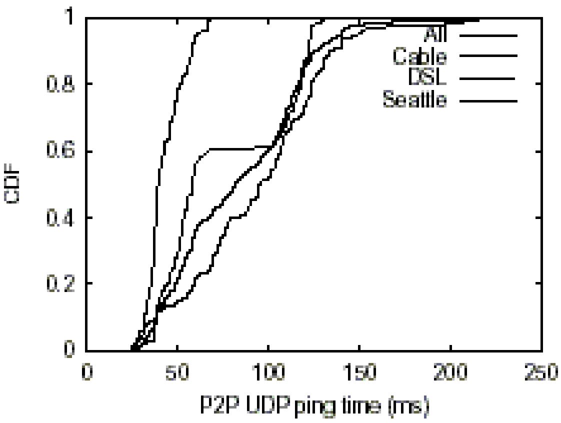


Figure 2 Peer-to-peer latency [4]

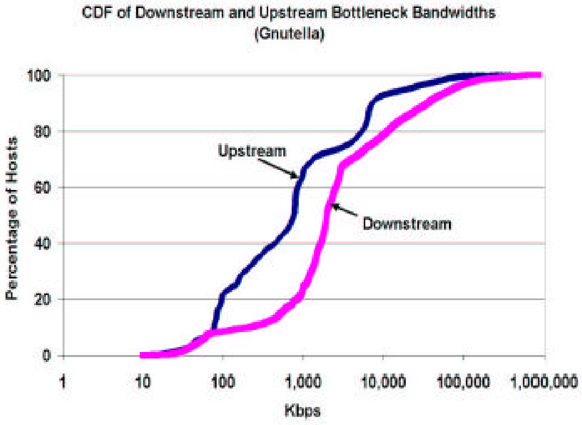


Figure 3 Upstream and downstream bottleneck bandwidths for Gnutella peers

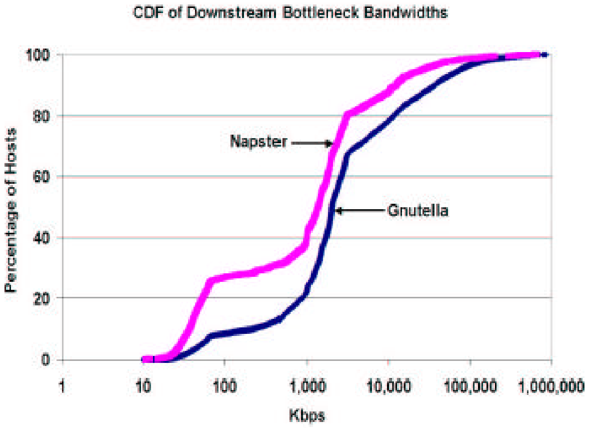


Figure 4 Downstream bottleneck bandwidths for Napster and Gnutella peers.

Figure 4 shows the CDF of downstream bottleneck bandwidths for both Gnutella and Napster according to [1]. The authors suspect that the larger bandwidths of Gnutella users can be explained by two factors. First, low bandwidth peers choose not to participate in the network because Gnutella´s flooding protocol would cause too much overhead for them. Second, users of Gnutella tend to be more technology-oriented and thus have higher bandwidth connections.

Figure 5 and Figure 6 present the bandwidth results according to [4]. In this paper, TCP throughput is measured directly, not approximated by bottleneck bandwidth as in [1]. This gives a more realistic evaluation of peer-to-peer throughput. In Figure 5 the CDF of TCP throughput is shown both in upstream and downstream direction in a case where the other end is always a well-connected server at Microsoft. As in [1], the results show significant asymmetry in the throughput which strengthens the observation that limited upstream throughput will most likely be a problem for peer-to-peer applications.

In Figure 6 TCP throughput is shown as measured directly between the peers. The median throughput for cable modem hosts is 220 Kbps while for DSL hosts the median is 120 Kbps. From this we can conclude that the correlation between TCP throughput and delay is quite weak (see Figure 2).

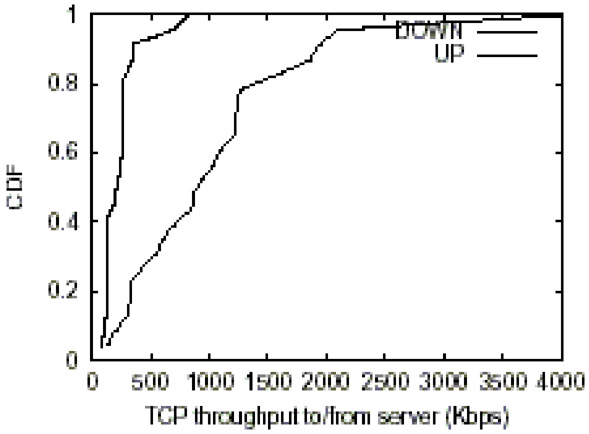


Figure 5 TCP throughput between the peers and a well-connected server [4]

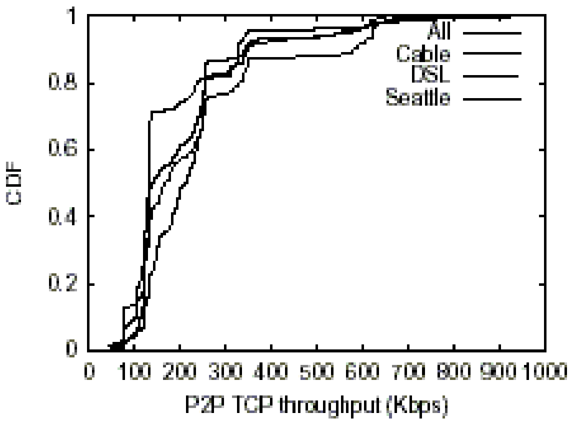


Figure 6 Peer-to-peer TCP throughput [4]

5.3 Availability

High availability is expected from hosts that serve content. In Figure 7 the CDFs of IP level and application level (Napster and Gnutella) uptimes are shown based on the measurements of [1]. In IP level uptimes the differences are minor. However, by looking at the application level uptimes it can be concluded that Napster peers tend to be available for longer times: when considering the best 20 % of peers, in Napster these peers have uptimes of minimum 83 % while in Gnutella the value is only 45 % [1]

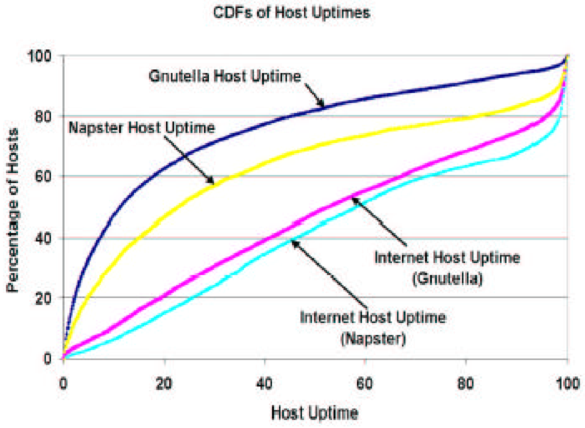


Figure 7 IP-level and application level uptime of peers [1]

Figure 8, Figure 9 and Figure 10 present the most important availability results from the measurements conducted in [2]. Figure 8 presents first the effect of the address aliasing phenomenon. As can be observed, host availability will be underestimated if the measurements are based on IP addresses. At worst, the factor of underestimation could be four. This result indicates that address aliasing should be taken into account in future measurements, since up to the present nearly all measurement studies have based on using IP addresses.

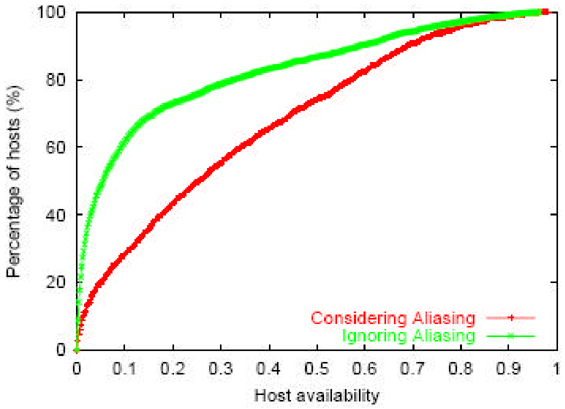


Figure 8 Host availability: host ID probes vs. IP address probes [2]

In Figure 9 the availability distribution is shown for three different averaging intervals: 10 hours, 4 days and 7 days. With longer periods the availability decreases since it is more likely that the host will become unavailable during that period. Thus, distributions of node uptimes (used e.g. in [1]) provide a more consistent view of availability since those results are not affected by the averaging period.

Figure 10 depicts the time-of-day effects in host availability. The x-axis represents the local time, with ticks indicating the dates at midnight. It can be observed that there is a clear diurnal pattern: availability is highest in the afternoon and in the evening.

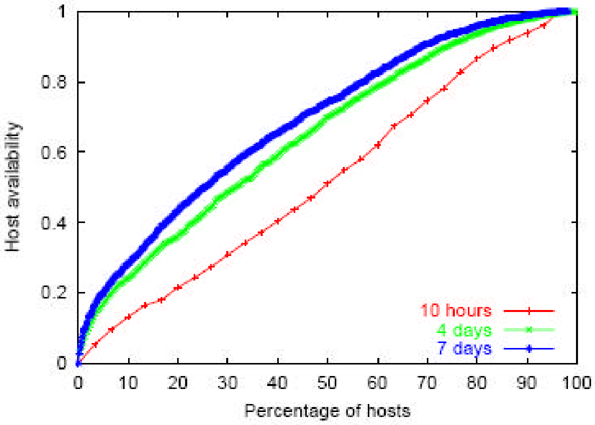


Figure 9 Availability distribution over different time periods [2]

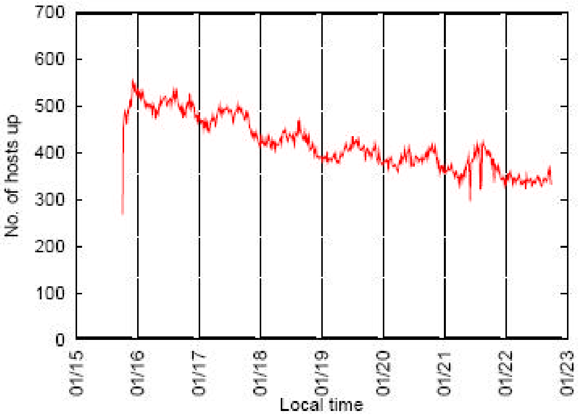


Figure 10 Diurnal patterns in number of available hosts [2]

5.4 Query hit ratios

In [20] it is shown with the help of simulations that query hit ratio depends much on the search method used. With "blind" search, where no information is available about which peer may have the desired content, hit ratios are as low as 40-50 % while with more informed search methods hit ratios above 90 % can be achieved. However, informed methods require indices to be maintained and updated when hosts join and leave the network.

5.5 General properties of peer-to-peer systems

Ideally, all participants in a peer-to-peer system should operate both as a client and a server. However, according to the results of [1] and [4] it seems that in practice the peer-to-peer systems behave much like the ordinary client-server architecture. First, a large part of the peers are so called free riders that contribute only little data to the system but issue a significant amount of downloads. According to [1], in Gnutella 25% of the peers do not share files at all and also in Napster 50% of peers share only about 5-20% of all the files shared. The peers are also not willing to cooperate to the extent that would be expected: According to [1] 30% of users that report to have a bandwidth corresponding to a standard modem connection actually have higher bandwidths. Second, there is considerable heterogeneity in the bandwidth, latency and availability among the peers ([1], [4]), meaning that only a fraction of the peers fit in the profile of a server (low latency, high bandwidth and high availability). Thus, the peers in the system should not be valued equally.

6 Implications of measurement results on system development

6.1 Peer selection

When a peer issues a file query, it will receive a response containing a list of possible locations where the file could be downloaded. According to the measurement results presented in previous sections the peers differ in their suitability for serving content, meaning that the delegation of responsibilities among the peers should not be uniform. Thus we believe that proper peer selection algorithms could improve the performance of peer-to-peer systems significantly. In [4] the authors discuss what metrics could be used for peer selection in two different situations: when forming a peer-to-peer search network and when retrieving files. They suggest that in the search phase where quite short messages are transferred latency is an important metric. Thus some kind of delay-vector based estimation should be used for identifying close peers in terms of latency. Here latency refers only to network delay but in reality also processing delay affects the search. For file retrieval, which is in general very bandwidth intensive it is suggested that uplink bottleneck link bandwidth should be used as a first order approximation for peer selection, since latency tends to be a bad predictor of TCP throughput.

In [6] some peer selection strategies have been proposed for the content retrieval phase. These strategies have been divided to user algorithms and global algorithms. Global algorithms consider the state of the whole system when making decisions while user algorithms only require state information related to the request. Obviously, global algorithms could only be deployed in systems like Napster that use central servers able to collect global information. The most interesting user algorithms investigated in [6] for content retrieval phase are

- **Fastest link** This algorithm selects for downloading the peer that has the largest uplink bandwidth. It does not take into account the number of current sessions served by that peer.
- **Greedy** This algorithm selects a peer that has the maximum available bandwidth, i.e. it calculates the value b/(n+1), where b is the uplink bandwidth of the serving peer and n is the number of uploading sessions.

The authors have performed simple simulations with these algorithms. Their results show that the performance of the fastest link algorithm is in general bad, because all downloading peers tend to use the same fast peers regardless of their load level. In many situations a peer with smaller bandwidth would be more suitable if it does not have so many concurrent uploads. We believe that the greedy algorithm could be useful for peer selection in the file retrieval phase, but it would require an easy way to obtain the number of uploading sessions on each peer.

6.2 Replication

The measurement results showed that availability is a crucial problem in peer-to-peer systems. In order to alleviate this problem, some form of replication should be used: if data is copied to many nodes, then the probability of finding at least one of these copies is increased. Furthermore, replication also tends to decrease network load [8].

Another reason for using replication is that in distributed, unstructured peer-to-peer systems (such as Gnutella and Kazaa) network topology and location of data are unrelated. This implies that when a peer sends a file query, it does not know which host could best solve the query [7]. In peer selection the decision in the search phase was based on latency. However, the number of hops that must be traversed before locating the file is equally important. Replication can provide a significant improvement when the goal is to minimize the search size, i.e. the number of hosts that have to be probed before the requested file is found.

In [7] different replication strategies in unstructured peer-to-peer networks have been examined. The following important strategies have been identified:

- **Uniform replication** Uniform replication replicates everything equally. When a file enters the system for the first time the system creates a fixed number of copies.
- **Proportional replication** In the proportional strategy more popular files are replicated more often. Each time that a certain file is queried, a fixed number of copies is created.
- **Square-root replication** In square-root replication the ratio of the number of copies is the square root of the ratio of query rates.

The uniform and proportional replication strategies are the two extremes of replication. Square root replication lies between these strategies. According to [7] square root replication provides the best result in terms of query size. Besides the number of copies made, another important problem in replication is how the replicas should be located. However, this question is outside the scope of our paper.

6.3 Caching

According to [5] and [15] there is significant temporal locality in the queries of peer-to-peer systems, that is, many queries are submitted more than once. This implies that caching of query responses could provide bandwidth savings and reduce query response times. However, query caching in peer-to-peer systems is not as simple as web document caching or search engine query caching. This is because the peers may join and leave the network frequently, meaning that cached responses can become out-of-date soon. Also the coverage of a query's response depends on where the query was issued from [15].

In [15], an initial caching algorithm for Gnutella is proposed that takes into account different coverages of queries. Denote by C a Gnutella client and by N_i its neighbors. The basic idea of the algorithm is the following: When a neighbor N_2 sends a query to a client C, C checks if it has the query with the same text and TTL in its cache. If so, and if the previous query was also sent by N_2, C returns the response from the cache. However, if the query is found but it was previously sent by N_1, C forwards the query to N_1 and combines the results it receives with the results already in the cache. Finally, C forwards the combined results to N_2. According to the simulations conducted in [15] this caching method can reduce the amount of queries sent by a factor of two and requires only a few Mbytes of memory.

7 Conclusions

In this paper we studied performance measurements and availability problems in file sharing peer-to-peer systems. We identified that from the user´s point of view the most relevant performance parameters are the time required to locate the requested file and the rate at which the located content can be downloaded. In most of the previous papers these performance parameters have been measured indirectly by studying IP-level latency, bandwidth (either bottleneck bandwidth or TCP throughput) and availability of the peers.

We investigated what kind of measurement methodologies can be used for evaluating the performance of peer-to-peer systems. We found that up to the present the measurements have been conducted either from a fixed measurement point by sending packets to peers that are first discovered by crawlers or directly between volunteer peers that agree to use required measurement software in their computers. We also identified that special care has to be taken when the crawlers discover peers based on their IP-address: there may not be a one to one mapping between a peer and an IP-address due to DHCP and NAT boxes.

We presented several performance results from previous peer-to-peer system measurements. These results showed that there is large heterogeneity in the latencies among the peers and that many peers will suffer from high latency. The bandwidth results revealed significant asymmetry. This was expected as many peers use asymmetric access technology, such as ADSL or cable modem. It seems that limited upstream throughput could be a problem for peer-to-peer applications. This should be taken into account when designing new access technologies. However, symmetric access technology would not necessarily be the best solution since it could increase the amount of peer-to-peer traffic further. Availability results showed that peer-to-peer systems are highly dynamic and that availability is relatively low, especially during certain times of the day. Query efficiency results revealed that query hit ratio depends much on the search method used. With informed search methods hit ratios above 90 % can be achieved.

Based on the performance measurement results we discussed how peer-to-peer systems should be developed in the future. First, we identified that since peers differ in their suitability for serving content, peer selection algorithms that identify “server-like” peers should be used. Second, we proposed that efficient replication strategies could compensate for the low availability of peer-to-peer systems. Third, we stated that caches utilizing the temporal locality observed in peer-to-peer queries could result in faster response times and large bandwidth savings.

We believe that measurements are an important means for investigating performance issues of peer-to-peer systems. However, measuring large scale systems is very time consuming and requires an enormous amount of measurement software to be developed. Thus, we suggest that some kind of common measurement platform for peer-to-peer systems should be constructed. This platform could be used for performance evaluation of peer-to-peer systems and for data collection so that detailed enough models of peer-to-peer systems could be developed. These models could be utilized in developing simulation standards for peer-to-peer systems.

8 References

[1] Stefan Saroiu, P.Krishna Gummadi and Steven Gribble: A Measurement Study of Peer-to-Peer File Sharing Systems, MMCN 2002.

[2] Ranjita Bhagwan, Stefan Savage and Geoffrey Voelker: Understanding Availability, IPTPS 2003.

[3] Jacky Chu, Kevin Labonte and Brian Levine: Availability and Locality Measurements of Peer-to-Peer File Systems, ITCom: Scalability and Traffic Control in IP Networks II, volume 4868, July 2002.

[4] Karthik Lakshminarayanan and Venkata Padmanabhan: Network Performance of Broadband Hosts: Measurements and Implications. Technical Report (MSR-TR-2003-15), March 2003, Microsoft Research.

[5] Krishna Gummadi, Richard Dunn, Stefan Saroiu, Steven Gribble, Henry Levy and John Zahorjan: Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload, SOSP´03, October 2003.

[6] Li Zou, Ellen Zegura and Mostafa Ammar: The Effect of Peer Selection and Buffering Strategies on the Performance of Peer-to-Peer File Sharing Systems, MASCOTS 2002.

[7] Edit Cohen and Scott Shenker: Replication Strategies in Unstructured Peer-to-Peer Networks, SIGCOMM’02, August 2002.

[8] Kavitha Ranganathan, Adriana Iamnitchi and Ian Foster: Improving Data Availability through Dynamic Model-Driven Replication in Large Peer-to-Peer Communities, Global and Peer-to-peer Computing on Large Scale Distributed Systems Workshop, May 2002.

[9] Open Source Community. The free network project-rewiring the internet. In http://freenet.sourceforge.net/, 2001

[10] Open Source Community. Gnutella. In http://gnutella.wego.com/, 2001

[11] KaZaA file sharing network. KaZaA. In http://www.kazaa.com/, 2002

[12] Napster Inc. The napster homepage. In http://www.napster.com/, 2001

[13] Krishna Gummadi, Stefan Saroiu and Steven Gribble: King: Estimating Latency between

Arbitrary Internet End Hosts, SIGCOMM Internet Measurement Workshop 2002.

[14] Xinyan Zhang, Jiangchuan Liu, Qiang Zhang and Wenwu Zhu: gMeasure: A Group-based Network Performance Measurement Service for Peer-to-Peer Applications, IEEE Globecom 2002

[15] Evangelos Markatos: Tracing a large-scale Peer to Peer System: and hour in the life of Gnutella, 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid, 2002.

[16] Mario Schlosser, Tyson Condie and Sepandar Kamvar: Simulating a P2P File-Sharing Network, 1st Workshop on Semantics in Grid and P2P networks, 2003.

[17] Subhabrata, Sen and Jia Wang: Analyzing peer-to-peer traffic across large networks, Second Annual ACM Internet Measurement Workshop, November 2002.

[18] Alexandre Gerber, Joseph Houle, Han Nguyen, Matthew Roughan, Subhabrata Sen: P2P, the Gorilla in the Cable, Proceedings of National Cable & Telecommunications Association (NCTA), June 2003.

[19] Overnet website, http://www.overnet.com.

[20] Dimitrios Tsoumakos, Nick Roussepoulos: Adaptive Probabilistic Search for Peer-to-Peer Networks, P2P'03, September 2003.

Network and Content Request Modelling in a Peer-to-Peer System

Bai Xiaole
Networking Laboratory
Helsinki University of Technology
xbai@{cc.hut.fi, tct.hut.fi}

Abstract

Peer-to-Peer communication is a trend and there are many peer-to-peer applications. The fact that peers can find each other in a large network is explained by the Small world model. Especially, this applies to the pure distributed Peer-to-Peer systems. In this paper, a new connection establishment preference model, Time Shifting Zifp Distribution model, is proposed for situations of Small-World power-law scalability.

To support our model, we argue that one important shortcoming can be found in recent simulations for peer-to-peer applications, especially concerning file-sharing systems. The traditional simulations cannot give an objective view about the performance in practice since content popularity is not properly taken into account. We propose a way to improve these simulations and give an example.

We emphasize that this new preference model applies for cases of Small-World power-law scalability, and may not be limited to Peer-to-Peer applications although Peer-to-Peer systems are the focus in this seminar paper.

Keywords: Peer-to-Peer, network modelling, content locating and retrieving, Small-World, time shifting zipf distribution.

1 Introduction

1.1 Some Concepts

Paper [1] gives the definition for **Peer-to-Peer**: “A distributed network architecture may be called a Peer-to-Peer (P-to-P, P2P, …) network, if the participants share a part of their own hardware resources (processing power, storage capacity, network link capacity, printers, …).These shared resources are necessary to provide the Service and content offered by the network (e.g, file sharing or shared workspace for collaboration): They are accessible by other peers directly, without passing intermediary entities. The participants of such a network are thus resource (Service and content) providers as well as resource (Service and content) requestors ”. In short, the difference between a Peer-to-Peer networking system and a Server-Client networking system lies in the role nodes play.

Furthermore, Peer-to-Peer networks can be classified into two subclasses according to whether there are some centralized entities (or services) provided or not. One subclass is “**Pure Peer-to-Peer** ”, which is defined as “ A distributed network architecture has to be classified as a ‘pure’ Peer-to-Peer network, if it is firstly a Peer-to-Peer network and secondly if any single, arbitrary chosen Terminal Entity can be removed from the network without having the network suffering any loss of network service ”in [1]. The author also gives the definition for the other subclass “**Hybrid Peer-to-Peer**” as “A distributed network architecture has to be classified as a ‘Hybrid’ peer-to-peer network, if it is firstly a Peer-to-Peer network and secondly a centred entity is necessary to provide parts of the offered network services ”.

When compared with the Client-Server network architecture, a Peer-to-Peer system has the following advantages: 1) Inherent scalability that is especially clear in pure Peer-to-Peer architectures. 2) Availability of more Information. However, we also notice that in a Peer-to-Peer network it is very difficult to provide any guarantee of Quality of Service [3].

Although [1] gives us some clear definitions for Peer-to-Peer from the point of view of network element functionalities and network architecture, it is still necessary for us to distinguish three related concepts we often hear mentioned: Peer-to-Peer Computing [2], Peer-to-Peer Network and Peer-to-Peer communication (See Figure 1).

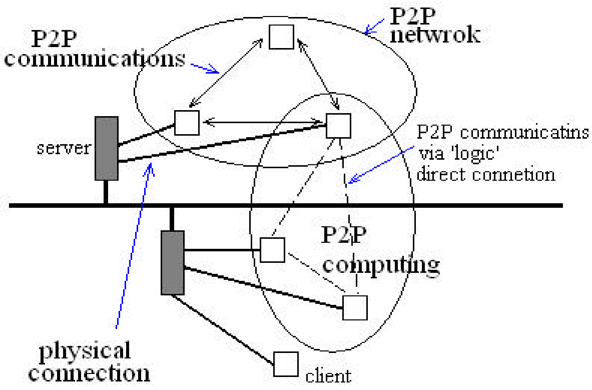


Figure 1 An illustration for three concepts

The concept of Peer-to-Peer network emphasizes more on the functionality blocks of each node, which make possible Peer-to-Peer communication between any two nodes in the network. However, Peer-to-Peer communication can also occur between two computers in a network that is not Peer-to-Peer. Peer-to-Peer communication enables both participants to initiate, manage, and terminate the session. Peer-to-Peer network refers to the physical network while Peer-to-Peer communication refers to one mode of communication. Different from Peer-to-Peer network and Peer-to-Peer communication, Peer-to-Peer computing defines an end-user application-level environment where Peer-to-Peer communication is only one way to facilitate the sharing of resources. Peer-to-Peer computing does not imply that every node can be a server to every other. In Peer-to-Peer computing the relationship between users is negotiated in some manner that is usually supported by some software layers providing server services.

1.2 Peer-to-Peer application areas

Peer-to-Peer computing has been around actually for several decades. The Internet conceived in 1969 was a Peer-to-Peer system. The goal of the original ARPANET was to share computing resources around the US. Prior to Napster's launch in 1999, one of the earliest experiment of large-scale Peer-to-Peer computing took place in 1994 when two scientist created a single clustered computer using 16 networked processors at the Goddard Space Flight Centre in Maryland.

Table 1 [4] shows the Peer-to-Peer application areas:

Area	Example
File sharing	Gnutella
Distributing Computing	SETU@home
P2P Search Engine	OpenCOLA
P2P Communication	ICQ
Edge Servers	Intel's upcoming edge server.
Device Intercommunication	Bluetooth
Anonymity/Anti-Censorship	Onion Routing

Table 1 P2P application areas

1.3 Organization of this paper

The rest of the paper is organized as follows: Section 2 gives an overview of Peer-to-Peer File Sharing architectures and some taxonomy. Section 3 introduces an important networking model, called the Small-World model, and its relationship with Peer-to-Peer systems, especially pure distributed Peer-to-Peer systems. After these preparations in Section 4, we propose a new connection establishment preference model, which in Peer-to-Peer file sharing can be considered as a model for content requests. Section 5 shows a way to implement it. After that, we use a modified Freenet simulator in our example. Based on traditional simulation results and results got from the new model, comparison and analysis are given. Section 6 concludes the papers. Section 7 outlines possible future work.

2 Peer-to-Peer File Sharing Systems Overview

2.1 Content (File) Sharing Architecture

There are three kinds of file sharing architectures in existing Peer-to-Peer applications.

2.1.1 Hybrid Centralized Peer-to-Peer File Sharing system

In such systems, some central unit facilitates the interaction between peers by maintaining directories of shared files stored on the respective PCs of registered users of the network.

A typical example is Napster [13], whose architecture is shown in Figure 2.

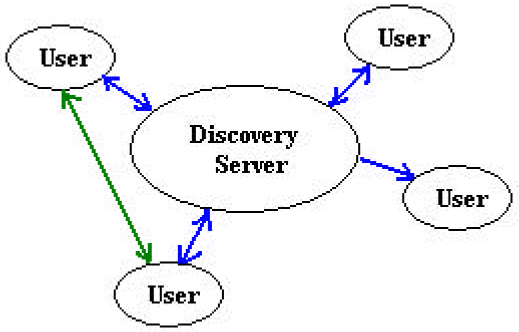


Figure 2 Central server provides directory

In the Napster architecture a central directory server maintains an index of all the files in the network. The metadata might include file names, creation dates, and copyright information. The server also can maintain a table of users' IP addresses and possible line speed. A query consists of a list of desired words. When the server receives a query, it searches for matches in its index. The query results including a list of users who hold the file are sent back to the user who initiated the query. The user then opens a direct connection with the peer that has the request file for downloading.

The advantages of a centralized indexing networking architecture are 1) locating files quickly and efficiently; 2) searching can be done as comprehensively as possible; 3) all users must register, which somewhat facilitates management and business.

The disadvantages are: 1) Vulnerability to technical failures; 2) The Slashdot effect: popular data becomes less accessible because of the load created by requests on the central server; 3) The central index might be out of date because the central server's database is only refreshed periodically.

2.1.2 Pure decentralized Peer-to-Peer File Sharing system

In such systems, peers may have different capacities but do have the same responsibilities. The communication between peers is symmetric. There is no central server index of the metadata of shared files that are stored locally on all peers (See Figure 3).

Typical applications are Gnutella [14] and Freenet [12][15].

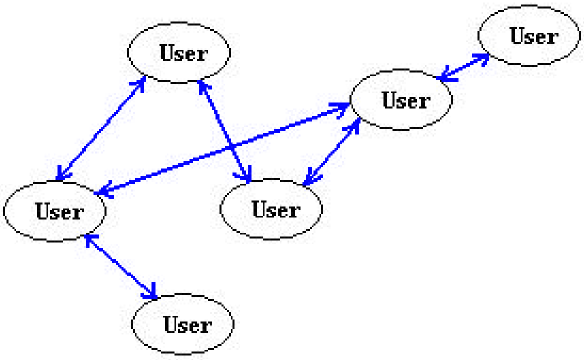
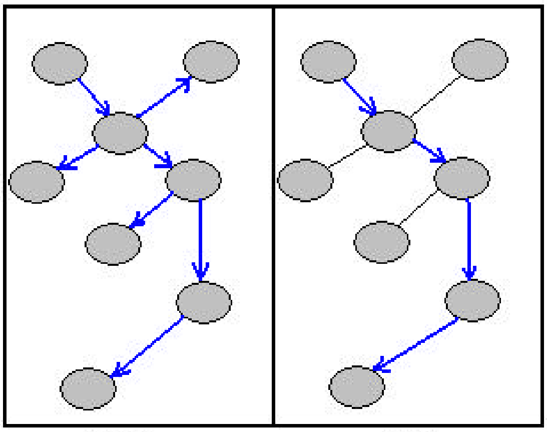


Fig. 3 No central server

Although Gnutella and Freenet belong to the same architecture class, they use different mechanisms to locate and retrieve shared files. The mechanisms in Gnutella are based on broadcast mode while those in Freenet are based on the chain routing mode. Fig.4 illustrates this.



query broadcasting *query routing*

Figure 4 Means to look for content

The most important advantage brought by the decentralized indexing network architecture is the inherent scalability. Additionally, fault tolerance is improved. The main disadvantages are slow information discovery and more query traffic on the network.

2.1.3 Partially Centralized Indexing System

Sometimes people include this kind of Peer-to-Peer system into the hybrid class. However, due to the big difference existing in the architecture, we classify them as a different class. Typical examples are KazaA[16] and Morpheus [17].

A central server registers the users to the system and facilitates the peer discovery process. After a Morpheus peer is authenticated to the server, the server provides it with the IP address and port (always 1214) of one or more "SuperNodes" to which the peer then connects. Local "SuperNodes" index the files shared by local peers that connected to it and proxy search requests on behalf of the peers.

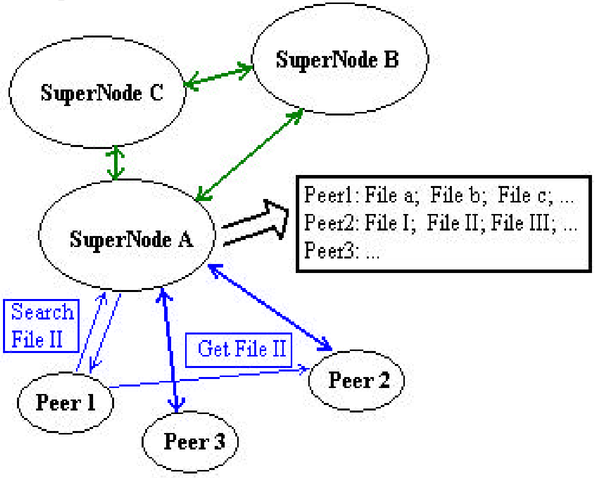


Figure 5 Local central server

As we can see clearly in Figure 5, this architecture compromises the characteristics existing in the former two network classes. Partial Centralized indexing network architecture reduces discovery time in comparison with purely decentralized indexing systems such as Gnutella and Freenet. It also reduces the workload of central services in comparison with a fully centralized indexing system such as Napster.

However, the way to choose "SuperNodes", local scalability, risk of out-dated data stored and effective search between "SuperNodes" are new problems brought by this compromise.

In fact, at the same time this architecture shares the advantages of pure centralized and pure decentralized indexing mechanisms, it has to face the disadvantages from both of them.

2.2 Another Way To Categorize

Let's look at some peer-to-peer systems that have appeared recently. Some mainly provide infrastructure for flexibly evolving the Internet, like CAN[18] and CHORD[19], some are mainly for large-scale network storage, like OceanStore[20]; some are mainly for multicast on the application level, like Yallcast[21] and some are mainly for anonymous publishing, like Freenet[12] mentioned above.

As core systems attracting an amount of attention, [18][19][20][12] are based on the hash-table mapping model. They essentially distribute <key, data> tuples across various nodes into a large network in a manner that facilitates scalable access to these tuples using the key. Hence, we can classify these systems into two categories: 1) the structured systems where the assignment of the key to the node on which the corresponding data is stored, is determined by the structure of the key space, and 2) the unstructured systems where no such assignment exits.
[18], [19] and [20] belong to the former and [12] belongs to the latter category.

3 Small-World Model

No matter which kind of taxonomy we use for Peer-to-Peer networks, three requirements are raised for content locating and retrieving. The first is effectiveness. The second is reliability and the third is scalability.
While policies, like key replacement in the cache or directory storage, play an important role in these requirements emerging from content locating and retrieving, a very important network model, the Small-World model, laying behind these policies, also plays a key role in the Peer-to-Peer network systems with distributed mechanisms, especially in the "pure" distributed systems.

3.1 What is Small-World

In many study fields, systems can be modelled as complex networks. The World Wide Web can be seen as a network of websites. The brain can be seen as a network of neurons. An organization can be seen as a network of people. We may find that the demand comes from many fields to explore the characteristics of a complex network itself and its dynamical behaviour.
Watts and Strogatz introduced the concept of Small-World network in 1998 when they were trying to make the transition from a regular lattice to a random graph [5]. Due to the rapid development of computerization of data acquisition and availability of high computing ability, huge databases on various real networks begin to emerge. That makes possible the exploration of properties of different kinds of complex networks. Two significant recent discoveries are the Small-World effect and the scale free feature of most complex networks.

3.1.1 Some Basic Concepts

Among many quantities and measures of complex networks, there are three spectacular concepts [6]. They are: average path length, clustering coefficient and degree distribution.

- ***Average Path Length***

In a network, the distance d_{ij} between two nodes, labelled i and j respectively, is defined as the number of edges along the shortest path connecting them. The diameter D of a network, therefore, is defined to be the maximal distance among all distances between any pair of nodes in the network. The *Average Path Length L* of the network, then, is defined as the mean distance between two nodes, average over all pairs of nodes.

- ***Clustering Coefficient***

Considering your friendship network, it is quite possible that your friends' friends are your direct friends. That is, two of your direct friends are quite possibly also friends.

Hence, very similarly, *Clustering Coefficient C* can be defined as the average fraction of pairs of neighbours of a node that are also neighbours of each other.

- ***Degree Distribution***

Maybe the simplest but also the most important characteristic of a single node is its degree. The degree k_i of a node i is usually defined to be the total number of its neighbouring connections. We also consider the importance of a node relative with its degree. The larger the degree, the "more important" the node is in a network. The average of k_i over all i is called the average degree of the network, and is often denoted by $<k>$.

3.1.2 Complex Network Models

Characterizing the topology of a complex network, we can usually get three kinds of network models.

- ***Regularly Coupled Networks***

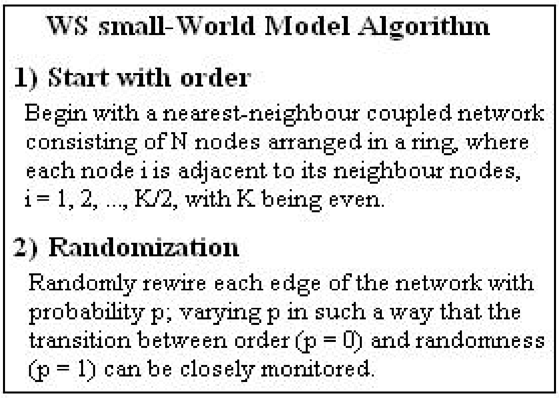
We call a network topology regularly coupled when the randomness of connection is got rid of. A widely studied regularly coupled network is called nearest-neighbour coupled network (a lattice), where every node is only joined by a few of its neighbours. The term "lattice" not only refers to a two-dimensional square grid but also to various geometries. A minimal lattice is a simple one-dimensional structure, like a row of people holding hands.

- ***Random Graphs***

At the opposite end of the spectrum from a completely regular network is the network with completely random connections, which were studied first by Erdös and Renyi (ER) about 40 years ago.

- ***Small-World Model***

Aiming to describe a transition from a regular lattice to a random graph, Watts and Strogatz [5] introduced an interesting Small-World network model, which is referred to as WS Small-World model. The WS model can be generated as follows:



WS small-World Model Algorithm

1) Start with order

Begin with a nearest-neighbour coupled network consisting of N nodes arranged in a ring, where each node i is adjacent to its neighbour nodes, i = 1, 2, ..., K/2, with K being even.

2) Randomization

Randomly rewire each edge of the network with probability p; varying p in such a way that the transition between order (p = 0) and randomness (p = 1) can be closely monitored.

The following Figure 6 shows the process more clearly:

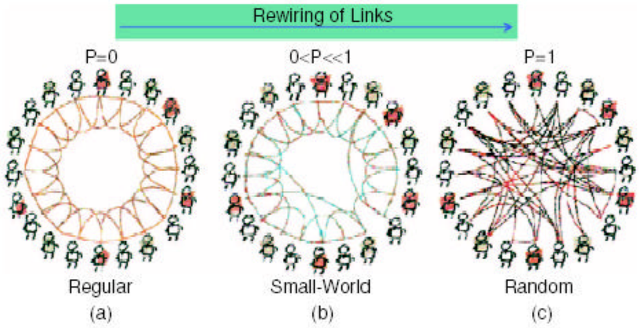


Figure 6 WS Small-World algorithm

As we can see in Figure 6 (a), it is a completely regular friendship network where each person has friends who are the nearest 4 to him. Fig.6 (b) shows a Small-World network, each people still knows 4 others on average but a few have some distant friends. Fig.6 (c) is a completely random network, everyone still knows 4 others on average but friends are scattered: few people have many friends in common and pairs are on average only a few degrees apart.

The Small-World network model is very important because of two reasons. First, investigation shows that many complex networks in real world are Small-World networks. Second, the Small-World has three significant characteristics: 1) relatively short average path length. This smallness infers the *Small-World effect*, hence the name of Small-World networks; 2) relatively high clustering coefficient and 3) they are scale free under power-law connectivity distribution.

3.2 Why Small World is important in Peer-to-Peer

End-hosts are becoming more ubiquitous, more powerful and more involved in service providing [9], which accelerates the development of new service and control mechanisms on the Internet. It is a trend that networks are evolving from the centralized client-server architecture to fully distributed architectures.

In any pure distributed Peer-to-Peer network system, like Freenet or ad-hoc network, Small-World property plays an important role due to the characteristics mentioned above.

3.2.1 Short Average Path length

It was an interesting discovery that the average path lengths of most real complex networks are relatively small, even in those cases where these kinds of networks have many fewer edges than a globally coupled network with an equal number of nodes.

Its meaning can be seen directly. In Peer-to-Peer file sharing or ad-hoc interaction, the sessions require low delay, which largely depends on the number of hops between the original node and the destination. The more relaying links, the more nodes or mapping processes the request will experience and thus the more likely the queuing delay plays a role.

Besides, based upon the dynamical changes happening all the time in the network, a long request delay will produce a larger possibility with which the destination or its content are unavailable when the request arrives.

In CAN [10], authors mention that the routing metric can be improved to better reflect the underlying IP topology by having each node measure the network-level round trip time, RRT, to each of its neighbours. This mechanism favours lower latency paths and helps the application level CAN routing avoid unnecessarily long hops. Mechanisms like the RTT-weighted routing, aim to reduce the latency of individual hops along the path. However, we notice that they do not utilize the Small-World effect since these mechanisms are localized while the Small-World effect gives us a global law. Without strong backup from Small-World effect, it is difficult to say that these local efforts are meaningful.

Furthermore, we may make our protocols simpler and more effective on a Small-World network.

3.2.2 High Clustering Coefficient

A Small-Work network shows a relatively high clustering coefficient, which in a simple example can be represented as a large fraction value for the fact that many of your direct friends are direct friends themselves. In Peer-to-Peer systems, it means that the nodes (or some contents) directly linked in one node are directly connected with each other with a high possibility.

What this characteristic can bring to the system is reliability and fault tolerance.

Let's take the file sharing in a pure Peer-to-Peer system as an example. Nodes are on and off from time to time and these procedures are hardly predictable. When a node gets off, not only its local sharing contents become unavailable, but also, e.g. in Freenet, its local hash table mapping is unusable affecting also correspondingly the content discovery route. Due to the relative high clustering coefficient, even when a node is down and its all direct links are removed in the network, there still exits ways for one node to contact others without large changes in average length of routes. Especially in some systems where requirements are set for some kinds of RTTs, this characteristic will provide search comprehensiveness in a dynamic situation.

Similarly, in ad-hoc networks, high level of clustering is also very helpful.

3.2.3 Scale Free

A significant recent discovery [11] in the field of complex networks is the observation that a number of large-scale complex networks including the Internet, WWW and metabolic networks are scale-free and their connectivity distributions have a power-law form.

Scale-free in pure distributed Peer-to-Peer network is also easy to understand. From time to time, some nodes can join the network and there is no limitation for the number of nodes. These new nodes bring new contents and therefore bring new links (key-location map) into the network.

Power-law degree distribution means that the probability that a node has k links has a power-law tail for large k, following $P(k) \sim k^{-r}$, where $r > 0$. This breaks the fixed node number limitation in WS Small-World model and is a condition for a scale-free network with Small-World characteristic [12].

In [11], authors suggest that two main ingredients of self-organization of a network in a scale-free structure are growth and preferential attachment. It points to the facts that most networks continuously grow by the addition of new nodes and new connections, and these new nodes are preferentially attached to existing nodes, for example, those with large numbers of connections i.e. “rich get richer”.

In Peer-to-Peer file sharing network, connections or links are established due to the interactions of requesting and retrieving files. Those nodes that have more content and more map links locally will have more chance to be connected, which refers to a kind of connection preference.

However, although in [11] author present a function to show attachment preference, not any preference shows to be power law scaling free. Hence, in a Peer-to-Peer network, we are interested in the questions 1) if a global preference with some specific policy will produce a scale free network with Small-World characteristics and 2) what kind of preference model we can use.

4 Time Shifting Zipf Distribution

4.1 What is the Zipf distribution

Zipf's law, named after the Harvard linguistic professor George Kingsley Zipf (1902-1950), is the observation that frequency of occurrence of some event P, as a function of the rank i when the rank is determined by the above frequency of occurrence, is a power-law function $P_i \sim 1/i^a$ with the exponent a close to unity [8].

Zipf curves follow a straight line when plotted on a double-logarithmic diagram. Figure 7 [7] shows a simple dataset with 300 elements that follow a Zipf distribution. We note that the line connecting the datapoints is linear on the right diagram (with logarithmic scales on both axes).

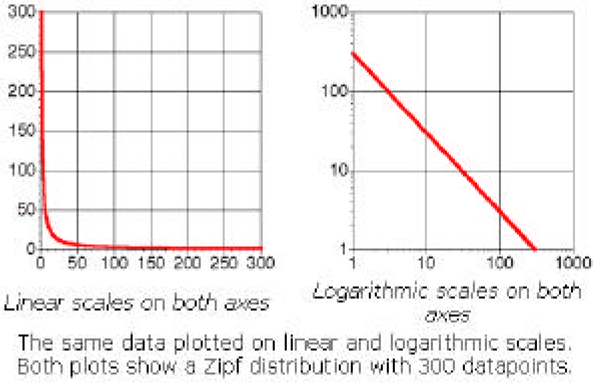


Figure 7 ZIPF distribution

There are several examples we can find about Zipf distribution. The most famous example of Zipf's law is the frequency of English words. See the statistic data for the top 50 words in 423 TIME magazine articles (total 245,412 occurrences of words), with "the" as the number one (appearing 15861 times), "of" as number two (appearing 7239 times), "to" as the number three (6331 times), etc. When the number of occurrences is plotted as the function of the rank (1, 2, 3, etc.), the functional form is a power-law function with exponent close to 1. The second example for Zipf is the population of cities (or population of communities). The population of the city plotted as a function of the rank (the most popular city is ranked number one, etc) also follows the power-law function with exponent close to 1. The popularity of books in a big library as a function of the rank is also an example of the Zipf's law.

A simple description of data that follow a Zipf distribution is that they have 1) a few elements that score very high (the left tail in Fig.7); 2) a medium number of elements with middle-of-the-road scores (the middle part of Fig.7); 3) a huge number of elements that score very low (the right tail in Fig.7).

4.2 Time Shifting Zipf Distribution

Every person tends to get resources he is interested in. This interest from a micro view varies from one people to the other along the time. However, we say there is some potential principle behind it, which is a time shifting zipf distribution.

As the concept of TSZD (time shifting zipf distribution) is first mentioned in this paper, next I will give a detailed explanation.

In each time snap, we notice the popularity of shared content follows the law of Zipf. As one example shown above, we can simply consider the popularity of content analogous to the popularity of books in some library. However, if we put this popularity on the time axis, we can see that the individual shifting character is clear. Some content is very popular during some period of time and then gets less and less popular. This shifting comes from the interest shifting of people. However, this shifting is relatively slow in an English text system because of the stable grammar system.

More accurately, the TSZD is a bridge between a macro view and a micro view. Shifting is an individual action while the Zipf is a global distribution.

In a Peer-to-Peer file sharing system, requests reflect current interest in some kind of content. For example, using eDonkey, people may usually look for new movies, or some good old movies. The new movies with strong advertisement or good reflection will attract more attention and thus their popularity will be in the left part in Fig.7 for some time. With new movies appearing continuously, requests may shift correspondingly.

In summary, the Zifp distribution shows global preference in discrete time snaps and TSZD shows both individual change and global expression.

5 TSZD in Peer-to-Peer Simulation

From above discussions, we know a Small-World network can bring us content location effectiveness and can improve the whole network content location reliability in peer-to-peer networks. These global effects surely make possible a simpler control and optimisation mechanism.

Furthermore, some kind of connection preference that in a peer-to-peer network is brought by content requests, may make a Small-World network scalable, i.e., allowing more nodes to join without changing the good network scalability characteristics.

However, from [5] we know that not all connection preference models fit into scalable Small-World networks. Due to TSZD representing the content requests in the real world more accurately, we can use TSZD to generate content requests in peer-to-peer network simulations and we can expect that the result is close to the real networks.

Simulations play a key role in [5] about power-law scalability of a Small-World network and they are also very important in peer-to-peer system design and performance evaluation [22].

However, up to now, all peer-to-peer simulations generate the content requests purely randomly, that is, every file is requested with the same possibility, which is not the case in the real world. Additionally, although simulations do take into account the node and connection changes with the time scale, time shifting of requests is not simulated correctly. In a short time scale, we should admit that time shifting of rank can be ignored due to the more significant effect brought by node and connection changes but the popularity distribution is still important. In a long time scale, the very "old" content will usually get less opportunities to be requested, that is, rank shifting happens and our content location policy should take this into account.

5.1 How to Implement TSZD

According to the zipf law principle "Rank×Frequency = Constant "[23]. Let P$_r$ be the Number of request occurrence of content of rank r over N where N is the total number of content request occurrences. For D unique contents, we get r×P$_r$= A, where A is a constant.

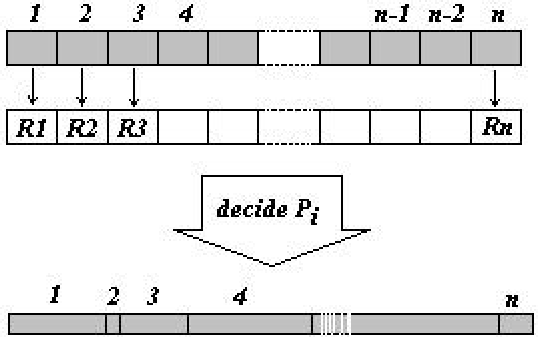


Fig.8 TSZD Implementation

We generate one set of random numbers first. Due to the randomness of these numbers, the rank list is decided for example the largest number can be ranked the first. This rank list is changing or shifting during the process when a new content is inserted into the network. The first number for today may not be the first for tomorrow.

When there are totally *n* numbers, we get

$$\sum_n R_i = k \; .$$

Given a constant A, it should be satisfied that

$$R_1P_1 = R_2P_2 = ... = R_nP_n = A$$

Now we get $P_i = A / R_i k$, however, because now

$$R_1P_1 = R_2P_2 = ... = R_nP_n = A' = A / k$$

we adjust A during the process according to value *k*. If *k* is fixed, it means that the content number is not changing and no adjustment for A should be done.

After we get P_i for each content *i*, we map the P_i into a linear scale as shown in Figure 8. Within the total scale, we generate random numbers with a uniform distribution. What content is requested is decided by where these random numbers appear in such a linear scale.

We now get the time shifting zipf distribution content requests and this implementation makes possible dynamic changes of the number of content items including both inserting and deleting. When the A' is given, the distribution of content requests at each time snap follows zipf distribution and from a long time scale view, TSZD is satisfied.

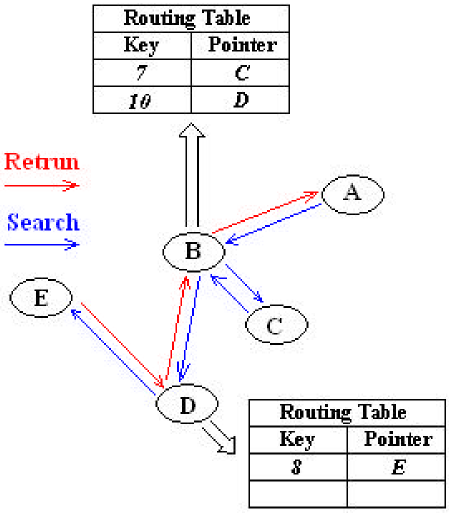
5.2 A Simulation Example

Due to the consideration that TSZD can more accurately simulate the real world content request in Peer-to-Peer systems, I modified the Freenet simulator you can download from [24] and made some comparison of simulation results.

Why I chose this simulator is because: 1) Freenet is of a pure distributed file sharing architecture and 2) this simulator shares a common interest field with this paper, i.e. it is mainly used to study content location and file retrieving performance.

5.2.1 Freenet content location and retrieving

Freenet [12] is a distributed anonymous information storage & retrieval system. In Freenet, files are identified by binary file keys obtained by applying a hash function to a string that describes the contents of the file. For this reason, we use the words *key, file,* and *data* interchangeably in this paper. Each node maintains a routing table that is a set of < *key, pointer* > pairs, where *pointer* points to a node that has a copy of the file associated with *key*. A steepest-ascent hill-climbing search with backtracking is used to locate a document. Loop detection and a HopsToLive (Freenet's TTL) counter are added to this basic scheme to avoid request looping and exhaustive searching.



Routing Table	
Key	Pointer
7	C
10	D

Routing Table	
Key	Pointer
8	E

Figure 9 Request routing in Freenet

Figure 9 shows a typical sequence of request messages. A request for key 8 is initiated at node A. Node A forwards the request to node B, which forwards it to node C since in B's routing table C is the node who has the closest key to key 8. Node C is unable to contact any other nodes and returns a backtracking "request failed" message to B. Node B then tries its second choice, D. Node D finds key 8 in its routing table and forwards the request to the corresponding node - E. The data is returned from E via D and B back to A, which ends this request sequence. The data is cached on D, B and A. An entry for key 8 is also created in the routing tables of D, B and A. Data inserts follow a similar strategy to requests [12].

In addition to the routing table, each Freenet node has a Data Storage Space (DSS). When a file is ultimately returned (forwarded) for a successful retrieval (insertion), the node passes the data to the upstream (downstream) requester, caches the file in its own DSS and creates a new entry in its routing table associating the actual data source with the requested key. When a new file arrives (from either a new insert or a successful request) which would cause the DSS to exceed the designated size, the Least Recently Used (LRU) files are evicted in order until there is room. Routing table entries are also eventually replaced using an LRU policy as the table fills up. Cache replacement scheme decides which <*key, pointer*> pairs are put into the routing table by choosing the files to be cached and then generating the corresponding <*key, pointer*> pairs. Route replacement policy decides which <*key, pointer*> pairs are chosen to be deleted from the routing table when the routing table fills up. The size of the routing table is chosen with the intention that the entry for a file will be retained longer than the file itself.

5.2.2 Simulation Method

In this section, we illustrate the performance of Freenet under heavy load using a simple simulation. The duration of the simulation was 12,000 time steps and the network had 300 nodes. Each node had a DSS limit of 40 files and a routing table limit of 90 files. The initial topology of the system is a ring: each node has a pointer to two neighbors. This initial topology is imposed by Freenet routing tables and need not have any relation to the underlying physical Internet topology. Each request is limited to 40 hops. Each node randomly generates and inserts a key (i.e., a file) with probability K per time step in the first 200 time steps (K varies in the range [0.005, 0.2]). All insertions are stopped after time 200.

In traditional simulation, the content request is generated purely randomly, that is, the request is uniformly distributed over the whole scale of content number. Every content item has the same possibility to be requested. In our new model, requests are generated according to time shifting Zipf distribution.

5.2.3 Simulation results

With TSZD content requests, after simulation, we can get Figure10 that is very similar with that in [12], shown as Figure 11.

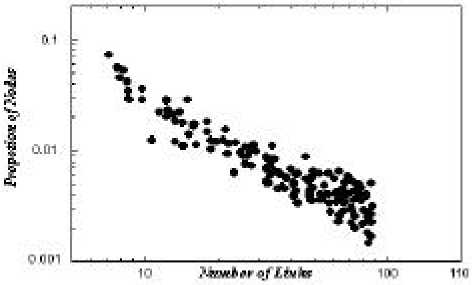


Fig. 10 Proportion of Nodes vs. Number of Links (in simulation where requests are generated with TSZD)

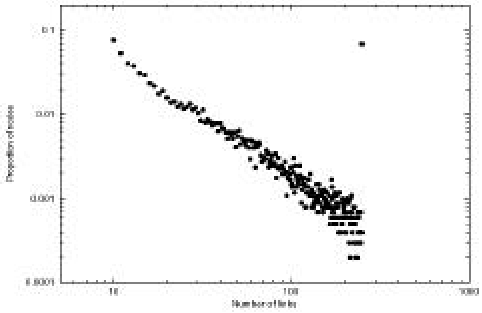


Fig. 11 Proportion of Nodes vs. Number of Links (in simulation where requests are generated randomly [12])

However, we have some different results shown in the following:

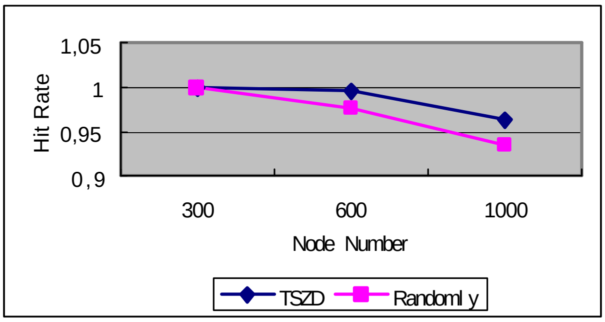


Fig. 12 Hit Rate vs. Network Scale

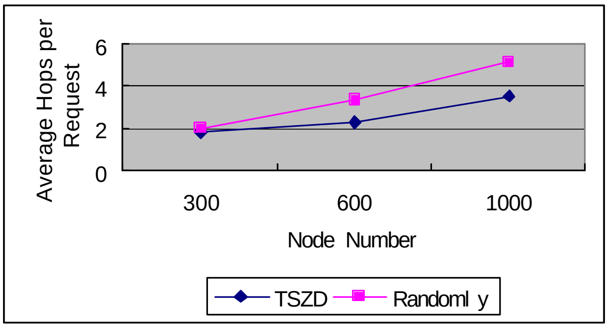


Fig. 13 Average Hops per Request vs. Network Scale

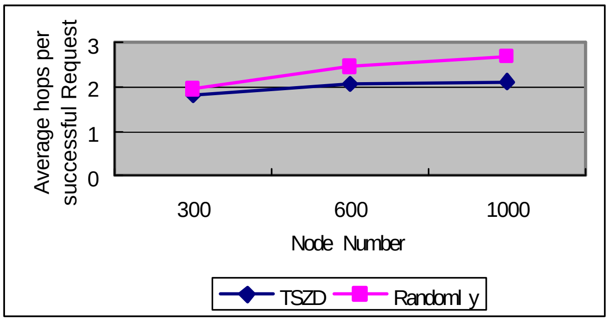


Fig. 14 Average Hops per Successful Request vs. Network Scale

We can see in Figures 12-14, how the Hit Rate, the Average Hops per Request and the Average Hops per successful Request show clear differences between the two requests generation laws. The Hit Rate is equal to the ratio of successful content requests to the total content requests. The Average Hops (per Request or per Successful Request) are the total hops (for total or successful requests) averaged on total number of requests.

The trend that the differences are getting larger along with the scale of the simulated network is clear.

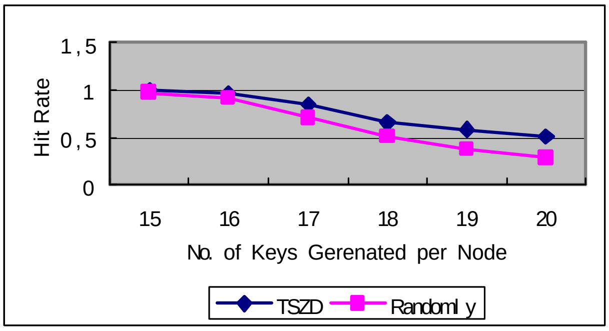


Fig. 15 Hit Rate vs. No. of Keys Generated per Node

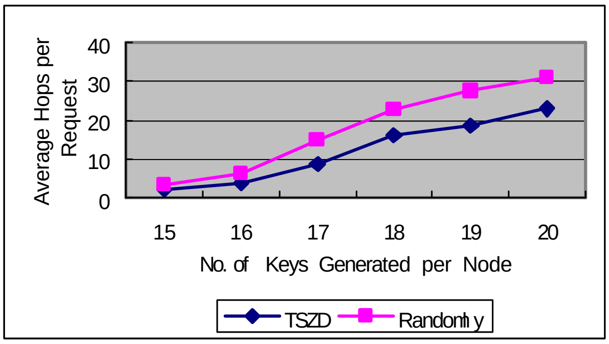


Fig. 16 Average Hops per Request vs. No. of Keys Generated per Node

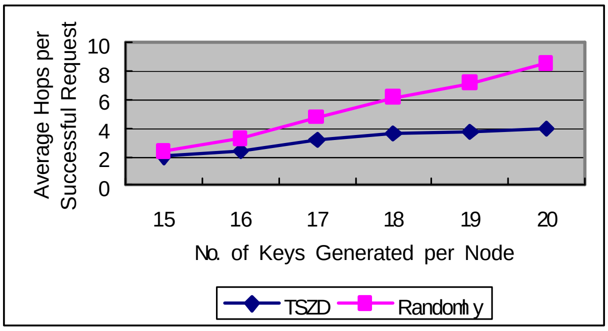


Fig. 17 Average Hops per Successful Request vs. No. of Keys Generated per Node

We also find that the differences are getting larger when the load of each node is getting heavier as we show in Figures 15-17.

5.2.4 Result Analysis

Clearly, using TSZD generation, we can get relatively better simulation results for Freenet than with requests distributed evenly over the content space. Of course, since we are doing the investigation of the simulation model, we do not care if results are “better” or “worse”. What is important is why the results from our new model are “consistent” or “different” when compared with those from previous work. We should also make a judgement about which simulation model is closer to the reality.

The consistent aspect is that Freenet shows Small-World characteristics [12] in both the traditional model and the new model, like we can see in Figures 10 and 11. Hence, we can say that the content location and retrieving policies in Freenet are quite good from the view of the whole network. That is to say, randomly or following the TSZD request generation both make a Small-World of routing entries. It will make possible effective content locating and retrieving and allows reliability and scalability through relatively less complex protocols.

However, in spite of the consistency in the macro view of the Small-World, differences in the two models do exit in some statistical parameters.

In Freenet, the main premise is that gradually the key space becomes more and more structured automatically due to the data request mechanism and the routing tables converge to a state where most of the queries are answered successfully and quickly. Freenet pays a lot of attention to anonymity and deniability, which is beyond the focus of this paper, and replication is a very integral part of the architecture.

Due to the replication mechanism, the data will be cached along the route as long as it is successfully found. When the request generation follows the time shifting zipf distribution, some contents are very often requested. These popular contents thus are cached in a large number of nodes, which facilitates next requests for the same contents. This fact can explain why the new model produces better simulation results.

Does the new model reflect the real world more accurately? Our answer is yes. We argue that TSZD is nearer to the practical request generation in a Peer-to-Peer file sharing system. Suppose we all share movies and some movies are very popular during some time. We note that, this time scale when some movies are very popular is relatively long when compared with the time scale during which computers join and leave. The requests generated during such a long time scale are of general interest no matter how many computers are on line. Besides, popularity seldom changes dramatically. Fashions usually fade gradually. Hence, content request generation following the TSZD model is closer to the real world.

6 Conclusion

Before we draw any conclusions, it should be emphasized that Freenet is only an example, we do not intend to investigate anything related to a special application.

A good Peer-to-Peer network can be modelled as a Small-World that can provide effectiveness, reliability and scalability in nature. Through studying content retrieving in Peer-to-Peer systems we give a new model for preference in connection establishment. The time shifting Zipf distribution, reflecting one kind of popularity distributions and their changes over time in the real world, actually can be widely used.

From an example, we found simulations with the new model may generate different results from those with the traditional model. We should pay attention to it especially when we use certain parameters as our judgement criteria on performance.

Let’s consider beyond the scope of this paper, Peer-to-Peer networking and content requests, Small-World network modelling can be suitable for many large complex distributed networks making further study in this field necessary. Also Time Shifting Zipf Distribution may be applied for connection preference beyond peer-to-peer networks.

7 Future Work

Some study may continue on the effect of connection preference in Small-World networks. Also the node’s (or connection’s) death and birth processes and distribution seems very interesting. I do believe they are very important and deep problems for any distributed control, dynamic and scalable networks.

References

[1] Schollmeier, “A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications”, IEEE, 2002

[2] David Barkai, “Peer-to-Peer Computing”, Intel Press, 2001

[3] Sean McCarthy, “Survey on P2P File Sharing System”

[4] “Peer-to-Peer Application Report”, http://netwserver.cerc. wvu.edu/classes/cs49h

[5] D.J.Watts and S.H.Strogatz, “Collective dynamics of ‘small world’ networks”, Nature, vol.393, pp. 440-442, June 1998

[6] Xiaofan Wang and Cuangrong Chen, “Complex networks: Small-world, Scale-free and Beyond”, IEEE Circuit and Systems Magazine, First Quarter, 2003

[7] http://www.useit.com/alertbox/zipf.html

[8] http://linkage.rockefeller.edu/wli/zipf/

[9] B. Maggs, Zhang Hui, etc, “Efficient content location using interest-based locality in peer-to-peer systems”, INFOCOM 2003.

[10] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, Scott Shenker, “A Scalable Content-Addressable Network”, Proceedings of ACM SIGCOMM 2001

[11] Albert-Laszlo, Reka Albert, “Emergence of Scaling in Random Networks”, Science, Vol 286, 1999

[12] Ian Clarke, Oskar Sandberg, etc, “Freenet: A Distributed Anonymous Information Storage and Retrieval System”, International Workshop on Design Issues in Anonymity and Unobservability, ed. By H. Federrath, Springer, NY, 2001

[13] Napster. http://www.napster.com

[14] Guntella. http://gnutella.wego.com

[15] FreeNet. http://freenet.sourceforge.com

[16]KazaA http://mrcorp.infosecwriters.com/p2p/p2p_study.htm

[17] “Teaching Materials Episode Four: Mystery of Morpheus”, http://reconstructors.rice.edu/recon1/lessons/Recon-TM4.pdf

[18] S. Ratnaswamy, P. Francis, M. Handley, R. Karp, and S. Shenker, “A scable content-addressable network”, ACM SIGCOMM, 2001

[19] I.Stoica, R. Morris, etc., “Chord: A peer-to-peer lookup service for internet applications”, ACM SIGCOMM, 2001

[20] D. Bindel, Y. Chen, etc., “Oceanstore: An architecture for global-scale persistent storage”, Proceedings of the Ninth international Conference on Architectural Support for Programming Languages and Operating systems, November 2000

[21] P. Francis, “Yallcast: Extending the internet multicast infrastructure”, http://www.yallcast.com/docs

[22] Hui Zhang, Ashish Goel, “Using the Small-World Model to Improve Freenet Performance”, IEEE INFCOM 2002

[23] Zipf’s Law, http://www.cpe.ku.ac.th/~arnon/Mirror/ir-p/ir2/sld002.htm

[24] Freenet simulator, http://netweb.usc.edu/~huizhang/freenet.html

Peer to Peer and ISPs´ Survival strategies

Aki Anttila
Cygate Networks
Vattuniemenkatu 21, 00210 Helsinki, Finland
aki.anttila@cygate.fi
11/2003

Abstract

The raise in peer-to-peer networking has been tremendous over the past four and a half years. It is estimated that currently at least half of the traffic in Internet is coming from P2P applications. It is not that there is no room for such a traffic. The real problem is that current user/usage profiles show that only a few percent of the users are using most of the bandwidth. This fact is especially important, when considering high-cost links, such as international transit connections or connections from an organization to an ISP. It is there, where extra traffic is not needed. This paper focuses on different possibilities to either fight against or live in peaceful co-existence with P2P traffic. There are multiple methods how these goals can be achieved. None of them stand out as a clear winner for an all-purpose P2P network machine. However, almost all of them could be used in live ISP networks to either advance user experience or to get rid of the ones that are hogging most of the bandwidth.

1 Introduction

After the original killer application for Internet – World Wide Web – another one has entered the arena in the past few years. This time it is not one that draws new users to the network but more a change for how Internet is used. This application – P2P networking – has been evolving into different branches with millions of users all around the world.

P2P networking is not a single entity. It consists of tens of different applications [1]. These can be divided into multiple subcategories. Three most famous are:

- FastTrack Network (with KaZaA and Grokster as clients)
- Gnutella Network (Morpheus, Bearshare, LimeWire, Gnutella etc.)
- OpenNap Network (over 50 client programs)

There are no credible estimates of how many users could be using these systems at least once in a while, but according to [2], in Q12003, most popular P2P clients were downloaded 79 million times and there were over 165 million registered users in top systems. Whether this number is increasing or decreasing is an interesting question. It seems that RIAA´s (Recording Industry Association of America) campaign against P2P piracy is paying off at least to some extent. The number of US households that were downloading pirated content through P2P systems is decreasing, and actually, consumers are deleting illegal content from their hard drives in a fear of a lawsuit, as market research company NPD shows [3]. However, this has not changed how often people are downloading latest versions of popular P2P client software. For example, Kazaa´s web page [4] says that there were 2,8 million downloads last week.

Although P2P networking has endless possibilities in gathering together people with mutual interests and allowing digital information sharing among them, it seems that the technology is mainly used for distributing pirated content. Most of the traffic that is distributed via P2P is music or movie files. It has not been researched, which percentage of these is actually pirated content and which is copyright free, but since RIAA and other copyright interest groups have felt an urge to fight against P2P systems, the percentage of pirated content must be high.

The evolution and pandemia-kind of distribution for P2P system usage has led network operators into a new kind of challenge. On the one hand they are struggling to find customers for their broadband services and to get traffic into their backbones (see [5]) and on the other hand they try to avoid using too much of their precious external network capacity (ie. transit links to other networks) to P2P packet transmission which, as we will see later on, has light-tailed usage distribution.

The rest of the paper concentrates on the network operator point of view. Section 2 shows measurements of P2P traffic from several recent P2P traffic analyses. Section 3 goes trough different approaches that network operators have for P2P traffic and Section 4 is a summary.

2 P2P traffic characteristics

Although P2P traffic has recently become somewhat problematic for network operators, there are not many studies available that analyze P2P traffic that flows in the wires. One of the main reasons for this could be the problems in capturing the massive set of data what such analysis requires. It is not that there are not powerful enough computers but they tend to cost a significant amount of money. Also, it is quite difficult to get access to an ISP network to monitor all the traffic that goes out and comes in. However, there have been co-operating operators and especially other organizations that have let researchers to capture network traffic and analyze it.

The main approaches for P2P traffic analysis in recent papers are:

- Gather e.g. Cisco´s NetFlow [6] information and analyze IP flows. This technique is used by [7,8].
- Monitor all traffic that goes to international/national peers of an ISP or an organization. This approach is used by [9,10,11,12,13].
- Monitor all signalling traffic generated by P2P clients. This method was not used in papers that I selected for the basis of this paper. The reason is that this method does not disclose the true distribution of the traffic. It just shows, what files are actually wanted by P2P users.

Besides the actual traffic analysis, another research method is to map the P2P overlay networks. This is usually done with "crawler" software that starts somewhere and gradually detects some part of the hosts connected to a certain P2P network. This method is used e.g. in [14,15]. Since this method only shows P2P network maps, it is not higly applicable for the purposes of this paper. However, since [14,15] provide excellent oversight e.g. about bandwidth usages, they are referenced here.

Next, I will summarize the most important results from the papers sited above. These results are needed, when I go through the different possibilities that are left for an ISP to react to the growing usage of P2P networking.

2.1 Number of users

How many users are actually using P2P networking clients and therefore participating in these networks is a question that does not have a clear answer. If we look at this issue globally, first we have to have an estimation of the total number of Internet users. This can be found from [16]. With no large-scale errors, it seems that there are around 600 million people online (i.e. connected to the Internet) this year. To estimate P2P users from this population, we can use very non-scientific (albeit quite applicable) 10 percent rule as the lower bound and findings of [9] as the upper bound. Using these we have an estimate that there are at least 60 million and at most 192 million P2P users worldwide. The latter figure goes quite well in line with [2] and therefore with large degree of trustworthiness, it can be said that there are 150-200 million P2P users worldwide. In other words, this means that every third or fourth user is using P2P applications.

Just out of curiosity, we can do the math for the Finnish ISP markets. Here I will concentrate only on broadband providers even, if there are users using also modems or ISDN connections. Based on the subscriber figures from [17], there was 344 000 BB subscribers in 7/2003. These are divided as follows:

- TeliaSonera has 110 000 subscribers,
- Elisa has 107 000 subscribers,
- Finnet Group has 77 000 subscribers,
- Others have 50 000 subscribers.

Of these, the number of P2P application users is (using one out of four rule):

- TeliaSonera has 27 500,
- Elisa has 26 750,
- Finnet Group has 19 250,
- Others have 12 500

We will come back to these figures later on in this paper.

2.2 Capacity of the users

Next we will look at, what kind of connections the users have. This is an important issue for two reasons. First, capacity reported by the user is used to select, from whom to download content. Another reason is that connection speed also gives an idea of how much a user generates load to ISP´s network.

According to [15], users are measured to have connection speeds as follows:

- 20-30 % are connected with more than 3 Mbit/s,
- 8-25 % are connected with less than 64 kbit/s,
- the rest, i.e. 45-62 % of users, are using connection speeds between these figures.

The measurement was done for Napster and Gnutella peers.

If we take the number of P2P users from 2.1. and use average percentages and capacity for user connectivity, we end up that there is a potential of generating huge loads of traffic to even Finnish provider networks. For the groups in question, this would mean an upper bound of:

- TeliaSonera 25,0 Gbit/s
- Elisa 24,4 Gbit/s
- Finnet 17,5 Gbit/s

- Others 11,4 Gbit/s

However, since these are BB subscribers, more accurate appraisal would be to estimate average connection speed and use that for upper bound. If we assume that average BB connection is 512/512 kbit/s (download/upload), then the upper bound for traffic flowing in one direction is:

- TeliaSonera 14,1 Gbit/s
- Elisa 13,7 Gbit/s
- Finnet 9,9 Gbit/s
- Others 6,4 Gbit/s

Lower bound for the traffic generated by P2P is estimated in [14]. It states that e.g. for Gnutella, basic signalling traffic is one gigabit per second for a 50 000 user network. In the light of Finnish broadband operators, this would be something like:

- TeliaSonera 550 Mbit/s
- Elisa 535 Mbit/s
- Finnet 385 Mbit/s
- Others 250 Mbit/s

This lower bound is pure signalling, no file transfers!

Various papers [e.g. 5] estimate that approximately half of all Internet traffic is P2P. And also that up to 70-80 % of P2P traffic is external to ISP own network [8]. This would mean that of the total 1,72 Gbit/s which is what P2P users MUST generate, about 1,3 Gbit/s is external.

Now, if we look at the traffic statistics in Ficix[2] [17], we find that the overall traffic through the exchange points is 1,9-3,8 Gbit/s. This goes (with small approximate errors) quite well in line with our previous calculations and the fact that half of the Internet traffic is P2P. For the obvious differences in these figures, it must be remembered that each of out BB operators have other external connections than Ficix as well. And – so far there have been no studies that would show what is the traffic distribution between internal, domestic and transit categories.

2.3 Content characterization

P2P systems are exchanging files that are totally different than what ISPs are used to. Traditional widespread applications like e-mail, news and WWW surfing have exchanged files (or data) that are fairly modest in size. Let´s say from few kilobytes to a few megabytes.

In P2P networks this is totally different. Most exchanged file types and their sizes are [10]:

- pictures (100-1000 kB)
- music files (2-5 MB)
- applications (50-150 MB)
- movie clips (50-150 MB)
- whole movies (600-900 MB)

There are no good estimations, how many unique files are available in P2P systems. However, taken from [10,11,12] it seems that we are talking about less than a million objects. To be precise, the measurements have recorded 100 000 – 600 000 unique objects. Even, if we take the minimum number from the measurements, this would be a huge number of unique files. To have some sort of idea of the amount of data there is, let´s use file size distributions from [13] and calculate how much data 100 000 files would be with an average size for different files as above. The estimate we get for the total amount would be 22 TB. As shown in [18], the US Library of Congress has texts worth of about 20 TB. So we are discussing about huge amounts of data.

The interesting thing here is that only a very small amount of unique objects are generating most of the file download traffic in P2P networks. As shown in [13], 10% of all objects are getting 70% of the download requests. Also [10] shows that only 0,1 % of all files are using 50% of the bandwidth.

By comparing file sizes and downloaded bytes [10,11], we observe that 60% of traffic is generated by files that are larger than 200 MB. This means that movie downloading is creating most of the traffic in P2P systems. This is an important result to which we will come back later.

2.4 User distribution

As already said, Internet Service Providers should see P2P networking as a way to increase traffic loads in their networks and therefore try to get more money from consumers. However, the most disturbing issue in P2P traffic is that only a small fraction of all users are using most of the bandwidth.

According to [12], only 4 % of Kazaa P2P clients generated 50 % of all the traffic by this application. Another 16 % was generated by the next 4 % of users. The same applies also for Kazaa "server" (i.e. upload) traffic. Saroiu et.al found that only a few percent of P2P users are generating most of the upload traffic.

In [8] the authors state, that about three percent of the users are generating almost half of the traffic. This applies to P2P traffic and also to the overall amount of traffic. Next 30 % or so is generating almost the other half whereas the rest of the users (almost 70 %) are generating only small amounts of data (around few percentages).

What these figures mean is that even if all subscribers are paying more or less the same amount of money for their

[2] Editor's Note: Ficix carries the ISP to ISP traffic in Finland. Ficix members have their own International links.

Internet connection and bandwidth, only a few of them are using a significant portion of network resources. If this usage would be purely inside the network of an ISP (i.e. the P2P overlay network would be connected more locally), then this would not pose a serious problem. ISPs have, however, sold Internet capacity to the subscriber and it would be not worth of money, if you could not use your access speed even when traffic stays inside the ISP´s network. Therefore, the problem lies in the external connections. To be even more precise – the transit connections to higher-level ISPs or backbone service providers. We assume that the same ratio of traffic usage by P2P clients applies also to external and transit connections and as a result too few users are using too much high value resources. One must remember that international capacity costs around 150 € per megabit/s per month whereas e.g. Ficix connection is 700 € per month for a two times Gigabit Ethernet connection.

That, only a handful of users are using a significant portion of the total external network capacity, is an important finding to which we will come back later.

3 Survival strategies

By now, it should be clear that P2P applications are creating a lot of traffic to ISP networks. By letting the pipes fill up, ISPs might encounter several issues that affect the service they are delivering to their customers. The most intriguing are:

- Network latencies increase, which might cause interruptions in interactive network applications. For example, online gaming might encounter sluggy responsiveness from other parties. This could render e.g. a war simulation game unusable.
- Packet loss might increase. Whilst P2P systems users do not notice (almost any) packet losses [see 11], again interactive or close to interactive (like web surfing) applications might suffer.

To equalize the amount of resources given to all of it´s customers, an ISP has a variety of choices that can be used either to enchance P2P usage or to fight against it. Following is a list of these possibilities (some of these can also be found from [9].

- Do nothing,
- Upgrade capacity,
- Ban P2P usage,
- Tiered services i.e. pay-per-use,
- Bandwidth limitations,
- Traffic limitations,
- Caching,
- P2P Redirection,
- Implement superpeers inside own network.

Next we will detail each one of these and also evaluate, the pros and cons of each of them.

3.1 Do-nothing model

Do-nothing model fits for the most profitable ISPs. If business is going well, there is enough capacity in the network and users are happy, why on earth rush into different kinds of tedious techniques to limit the traffic generated by one (albeit large) application?

Even if it looks a naïve approach to P2P problems, this is the one that is chosen by a large number of ISPs. At least, this is the case so far. Maybe their ideology is that "time takes care of this problem as well" meaning that maybe RIAA or other interest groups gain victories in their battle against online piracy or that network capacity grows so much anyway, that there is no need to restrict (or even think of it!) any traffic.

Hand-in-hand with do-nothing model often goes total unawareness of what is actually happening in the networks. As stated in [19], one of the major problems in the Internet is that there are too many network operators that don´t have a clue of what is actually going on in their networks. They don´t have even the basic idea of e.g. traffic matrixes. Even for their external (and therefore costly) bandwidth.

3.2 Upgrade capacity

The next possibility for ISPs is to upgrade their external capacity. Although, this might seem like an easy task, it might lead to several problems. Most important of these is that if more users discover that there is available external capacity to fill up, they might start using it. For the poor operator this means that they will constantly need to upgrade their external links. Over time the cost of the external capacity might ruin an otherwise profitable ISP business.

Another problem the ISP might run into is that its whole network structure cannot handle the load that should be flowing in/out from/of the external link. This means that the core network of the ISP does not use fat enough pipes to handle the extra traffic. This piece of the infrastructure is even more costly to upgrade just because of some bandwidth-hungry users.

Whilst upgrading capacity is an easy way to a) keep the existing customers happy and b) provide enough resources for the P2P users, it does not seem a rational thing to do. This is proved by simple calculus. Let us take the existing user base of e.g. Finnet group (roughly 20 000) P2P users and their estimated lower bound of usage (roughly 400 Mbit/s). Now, if we were to buy more external capacity to support existing customers´ added need for P2P file exchang for about 200 Mbit/s, which is a significant amount of external capacity especially if it is transit capacity to USA, the cost per user would be a couple euros per month. But given that the cost of a broadband connection is decreasing all the time, these two euros could collapse the whole revenue model. More problematic in this model is that this capacity upgrade advances the usage of 3-5 % of the whole user population. When the cost is divided among the high volume users, it is 35-50 euros per month. That is the cost of the whole residential DSL connection!

I presume that there are no operators that are using capacity upgrading as a measure to give all users adequate bandwidth inside their networks not to mention towards external networks when facing P2P traffic problems. It is quite obvious that this kind of an ISP would collect all the heavy-hitters from other ISPs and they could use all the bandwidth that is bought for them.

3.3 Banning P2P usage

At it´s earliest incarnations, P2P traffic was quite easily detectable from the rest of the network traffic. P2P clients behaved as network operators were used to expect from a TCP/IP application; they used well-known ports to originate TCP sessions. For example, Kazaa was running on port number 1214.

For the dismay of network operators this has changed. As illustrated in [20], a technique called port-hopping has found it´s way into P2P clients. For example, Kazaa version 2.0 was released in September 2002 and right after that traffic analysises showed two results. First, the ratio of recognized P2P traffic versus overall traffic decreased. At the same time the ratio of unrecognized TCP traffic versus overall traffic increased. This significant change shows how adaptable P2P clients are in changing environments.

However difficult it is, current P2P client traffic is traceable. Different vendors have implemented tools that look for certain kinds of signatures inside P2P traffic.

How does all this relate to the subtitle of this section? Well – if you are about to ban something, you must be able to enforce your rulings. Therefore, what operators could do is to ban P2P usage. And then disconnect the breachers of the rule from their networks. Actually, this has been done e.g. in University of Florida [21]. There this method has been highly successful. But then again, the users of this network have no alternatives.

How would banning of P2P usage affect the customers of a normal ISP? Well – it can easily be said that at this time 65-75 % of the customers would be happy. Since they are not using P2P currently, they would have more bandwidth for other purposes like e.g. web surfing. But the rest of the subscribers would propably seek for another service provider – one that has a positive attitude towards P2P networking.

One must also remember that currently P2P is used primarily for music/video downloads. What if and when this is qoing to change? Other possible usages of P2P technology is sharing personal digital recordings, be them songs, home videos or pictures, sharing computing resources or creating a new telephony network on top of Internet. There is a wide variety of possibilities for P2P networking. In the near-term future, who can say what can or cannot be done with P2P? If the usage of P2P techniques is commoditized, then an operator that bans using P2P must be insane since it would lose all it´s customers.

So far there has not been any (commercial) operator that I know of, who has totally banned P2P traffic from their networks. However, as illustrated in [21], there are multiple organizations, primarily universities, that have tried to get rid of P2P by banning it. The same phenomenon is also spreading towards common enterprises; although acceptable usage policy should strictly forbid using employers computing resources to non-business, P2P has found it´s way to companies as well. Therefore, more and more companies are explicitly banning P2P with severe consequences for the misbehavers. There is also a reason for it, since besides the fact that P2P is not a tool to do work, it can be used to spread viruses or even turn PCs into something else [see 22].

3.4 Tiered services

Tiered services means that instead of charging Internet customers a flat monthly fee, they would be charged per their usage. This model is a strong candidate in ISPs toolbox to fight against P2P applications and is used by quite many operators in the world.

Obvious downside of tiered services is that user churn might increase a lot. However, this is an easy question to tackle. As was discussed in 2.4, most of the bandwidth is taken by only a few percent of the users. Therefore, just letting these users go would greatly benefit the ISP´s network health and therefore the perceived quality of service for the rest of the users would be significantly improved. And this would be done without jeopardizing the revenue model.

Take for example Elisa. It has around 100 000 customers. For these, an estimated 25 000 are P2P users. Of these, around 1000 users are the most active. As [8] suggests, these would generate over one terabyte of traffic each day. That is almost 100 Mbit/s! If we estimate again, let´s say that of this traffic, half is going to fill up transit capacity. The calculation yields that providing Internet service for this group of thousand people is costing 15 000 euros per month. That is 15 euros per month per user just for transit. Clearly this is a group of unprofitable customers.

If an operator would go to this kind of model, two questions arise; 1) Is the tiered service used both for domestic and international or only for international traffic and 2) which is the capacity limit that can be used without extra cost?

The answer for question one is easy. Since domestic capacity is extremely cheap compared to transit capacity, it makes no sense to use tiered services on that. However, it might be technically difficult to separate, what part of the traffic is going to a domestic peering point and what part to international pipes.

The other question is not so straightforward. It must be remembered that we don´t want to frighten off all the customers but only a small portion of them. Therefore, the monthly fee should allow the majority of users to use Internet as before. But low enough quota should be set scuh that the heavy-hitters would leave and find another ISP whose network to fill up. Suggesting a limit is beyond the scope of this paper but something between 50 MB and 1 GB per day might be enough.

3.5 Traffic limitations

Traffic limitation is actually more or less the same as tiered service. The only difference is that instead of paying more for extra capacity used, when user quota is full, network service is denied.

Although this is easier to set up than the previous model, there are obvious downsides as well. The most notable of these is that if the user cannot get any kind of connection, he/she will for sure change the ISP. This might be what the original ISP had in mind, but this effect needs to be recognised.

Traffic limitations or some sort of tiered services are used and will be used by many operators. They are a fairly easy way to fight againts P2P usage.

3.6 Bandwidth limitations

Bandwidth limiting seems to be similar than tiered service and traffic limitation but there is a fundamental difference. In tiered service and traffic limiting what is looked after, is the total usage of network resources over a certain period of time. Bandwidth limitations are continuously watching, how much capacity a single customer is using at that time.

As in tiered service/traffic limitation, there are a couple of interesting issues that need to be investigated. These are; what traffic is actually limited, where limitation is done, how limitation is enforced? Let´s take each of these in turn.

If bandwidth limiting is chosen as the way to go, then it needs to be considered, what traffic will actually be limited. Actually the technology used for limiting affects seriously, what can be done in terms of different traffic types. But since the whole problem is related to P2P traffic, then it should be limited.

For the next question, the answer would be; limit on external capacity, allow internal/domestic usage.

Last question is more problematic. ISP can either choose to use traditional techniques like Cisco´s MCQ [23] or use some start-up´s like Ellacoya´s [24] innovative products. The problem with the former is that currently there is no build-in way to separate customers and different traffic unless add-ons like Rommon´s [25] technique is used. The second approach, however, does not have this kind of drawbacks and therefore it and similar kinds from other vendors are widely used in operator networks throughout the world. Whilst this is not shown by any study, announcements in vendor´s web pages and market rumors show that usage of traffic restricting equipment is a fact.

3.7 Caching

All the different approaches discussed so far have been designed to resist the usage of P2P applications or at least minimize their effect on the average ISP customer. Next I will introduce three methods that are designed to enchance the usage of P2P inside an operator network but at the same time reduce the effects on external capacity.

First alternative to do that is caching. This means that a similar kind of caching server is build for P2P traffic as is widely used for ordinary web traffic. Caching P2P traffic seems to be a tempting alternative for four reasons. First, there are (currently) only a limited number of files that need to be cached. Even, if we would cache everything i.e. build a perfect cache, the total number of files would be less than a million. Second, only a small portion of files accessed is generating over 50 % of the traffic (see section 2.3). Third, there are enough new users so that even if P2P files are typically of type load-at-most once, a cache could deliver files to some population. Finally, P2P files are static enough so that they do not have to be refreshed every once in a while.

Caching could be very effective in large P2P networks. This issue has been studied e.g. in [11,13]. The former studies how cache would behave, if the number of users changes. Their conclusion is that even for very small user populations, caching would save 40-60 % of the bandwidth. What is interesting is that for a huge population like half a million, caching would drop capacity usage by over 80 %. The latter focuses on estimating caching potential when plotted against disk (cache) size and the amount of network traffic. The authors´ estimate is similar to the previous one in that cache byte-hit-rate might exceed 80 %. What is even more interesting is that even relatively modest caching server disk capacity is sufficient to support high byte-hit-rates. This phenomenon is getting even better if there is much traffic in the network. For a couple of hundred gigabytes disk size, caching can decrease network capacity demand by 35-65 %, depending on network traffic amounts.

Even if caching could solve the bandwidth problems and still keep users happy, there is a huge downside in using cache servers. That is; legal issues. There is no network administrator in the world that would believe that there is only legal content in P2P networks. If caching is implemented, then it would mean that the ISP stores all kinds of illegal content for their users. Practically speaking, the ISP doing so would be a distributor of illegal content.

Even if there is a risk that the ISP gets sued by e.g. RIAA, some have chosen to deploy cache servers in their networks [25]. It will be interesting to see, whether some copyright interest group will try to identify and sue them. A safe method for caching would be to implement a piece of software that could "see" what part of the content is legal and what is not. Of course this means that e.g. songs must contain some sort of watermark. Based on the marking, some would be cacheable and some not. This kind of categorization would surely affect the performance of the caching scheme but at least it would bring well-slept nights to the CEO of the ISP in question.

3.8 P2P redirection

P2P redirection means that all the signalling traffic generated by P2P clients inside an operator network is passed through a redirector server. This server will examine the traffic and then decide, would the request be satisfied best by going outside of the ISP network or could the requested content be found from local P2P clients. Since a redirection server needs to participate and interpret all client requests, it must a) be on the path towards the external world and b) be able to talk all or at least the most relevant P2P protocols.

P2P redirection is a technique that has not been researched or implemented. Due to it´s potential pros both in reduction of the bandwidth and in increasing customer satisfaction, it would be a good future research topic.

3.9 Superpeering inside ISP network

Superpeering is a kind of a reduced set of features what P2P redirection server could offer. This technique means that ISP deploys e.g. a Kazaa super-peer. Then it advertises the existence of this super-peer to all users in it´s network. When this is done and when/if all users connect to this super-peer, then the P2P network topology will reflect that of the physical network. This would reduce the amount of traffic that needs to be carried in external network connections.

Super-peering sounds like a fairly good alternative as a countermeasure against P2P traffic loads, but has one major obstacle. That is user trust. Do ISP´s users trust the operator to think the best of their users? Over time this is propably possible but current P2P users might be afraid of the thought that the operator sees what kind of files they are requesting. However, this is more a social question and the answer to this lies beyond the scope of this paper.

An additional thing that could attract users to ISP-based super-peers is that they could be equipped with additional features and services [27]. One service that could be provided is better searching capabilities for the available objects.

3.10 Summary of the methods

All the methods discussed in previous sections are illustrated in the following table. Even if this study does not concentrate on legal issues, I have listed also IPR friendliness into the table because I feel that it is currently a crucial part of the P2P traffic control method selection process.

Method/property	**User**	**ISP**	**IPR**
Do nothing	Yes	No	No
Upgrade capacity	Yes	No	No
Bannning P2P usage	No	Yes	Yes
Tiered service	No	Yes	Yes/no
Bandwidth limitation	No	Yes	Yes/no
Traffic limitation	No	Yes	Yes/no
Caching	Yes	Yes	No
P2P redirection	Yes	Yes	No
Superpeers	Yes	Yes	No

Table 3: Summary of methods, User = user friendly, ISP = ISP friendly, IPR = IPR friendly

As one can see from the table, none of the traffic control mechanisms discussed so far satisfies all needs. The ban for P2P usage is the most effective in terms of ISP and IPR friendliness. But banning P2P would propably mean 10-25 % customer churn, which is not what operators want – especially now in a situation where broadband access market cake is shared. On the other hand, caching, P2P redirection and super-peering seem to be both user and ISP friendly, but since they actually advance P2P usage, they are not IPR friendly.

4 Summary and conclusions

P2P is here to stay. Even if current usage of P2P networking, namely, pirated content exchange fades out at some point in the future, new bright P2P services are already partly in use and partly on drawing boards. Therefore, it is good to know, how an ISP can react to the challenge caused by P2P traffic.

In this paper, I studied possible methods for ISPs to either fight against or to minimize the effects of P2P networking in their networks. First a range of studies about P2P traffic analysis was presented and they were used to build up a real-world case for the Finnish broadband ISP market. To summarize this, there are three issues that characterize P2P traffic;

- It fills up networks, using about 50 % of available bandwidth.
- Only a handful of users are sending or receiving a significant portion of P2P traffic.
- Even if a wide variety of files and file sizes is available, most of the traffic is caused by a small group of extremely large objects, movies.

Next I presented different approaches for P2P traffic handling. Besides the first two, all of them have the same idea; to reduce capacity needed for external (transit) connections. Some of them do excellent job (like caching) and some just try to justify extra cost by charging it from the heavy-users.

However, it should be noted that none of the solutions is perfect. If the ISP in question does not have to care about IPR issues (which is the case at the times), then caching, P2P redirection and super-peering are the winners. Then again, if IPR is a problem for the ISP, some sort of traffic limitation/tiered service combination would fit best. Do-nothing and more capacity models are clearly loosers since the first will only guarantee congested networks and therefore highly displeased customers whilst the second might cost all the money the ISP can get from it´s subscribers.

5 References

[1] Ghosemajumder, S.: Advanced peer-based technology business models, 2002, http://shumans.com/p2p-business-models.pdf

[2] USA Today: File sharing is a hit, despite legal setbacks, 05/2003, http://overclockersclub.com/newscomment.php?article=EpkEuklFyVlErfpgMS&

[3] NDP Group Press Release: Consumers Delete Large Numbers of Digital Music Files From PC Hard Drives, 11/2003, http://www.npd.com/press/releases/press_031105.htm

[4] www.kazaa.com, accessed 11/13/2003

[5] Odlyzko, A.M.:Internet traffic growth: Sources and implications, 2003, http://www.dtc.umn.edu/~odlyzko/doc/itcom.internet.growth.pdf

[6] Cisco Systems: Netflow Services Solutions Guide, accessed 11/13/2003

[7] Sen, S. and Wang, J.: Analyzing peer-to-peer traffic across large networks, 11/2002, Proceedings of ACM SIGCOMM Internet Measurement Workshop

[8] Gerber, A., Houle, J., Nguyen, H., Roughan, M., Sen, S.: P2P, the Gorilla in the Cable, 2003, http://www.research.att.com/~sen/pub/p2pCable2003.final.pdf

[9] Morin, M.: Managing P2P Traffic on DOCSIS Networks, 2002, http://www.sandvine.com/solutions/pdfs/Managing_P2P_Traffic_on_DOCSIS_Networks.pdf

[10] Leibowitz, N., Ripeanu, M., Wierzbicki, A.: Deconstructing the Kazaa Network, 2003, 3rd IEEE Workshop on Internet Applications (WIAPP'03)

[11] Gummadi, K., Dunn, R., Saroiu, S., Gribble, S., Levy, H., Zahorjan, J.: Measurement, Modeling and Analysis of a Peer-to-Peer File-Sharing Workload, 2003, Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP-19)

[12] Saroiu, S, Gummadi, K., Dunn, R., Gribble, S., Levy, H.: An Analysis of Internet Content Delivery Systems, 2002, Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI)

[13] Leibowitz,, N., Bergman, A., Ben-Shaul, R., Shavit, A.: Are File Swapping Networks Cacheable? Characterizing P2P Traffic, 2002, 7th International Workshop on Web Content Caching and Distribution (WCW)

[14] Ripeanu, M., Foster, I.: Mapping the Gnutella Network: Macroscopic Properties of Large-Scale Peer-to-Peer Systems, 2002, IEEE Computing Journal

[15] Sariou, S., Gummadi, K., Gribble, S.: A Measurement Study of Peer-to-Peer File Sharing Systems, 2002, Proceedings of Multimedia Computing and Networking 2002 (MMCN '02)

[16] Online populations, accessed 11/13/2003, http://cyberatlas.internet.com/big_picture/geographics/article/0,1323,5911_151151,00.html

[17] www.ficix.fi, accessed 11/16/2003

[18] Odlyzko, A.: Content is not king, 2001, http://www.dtc.umn.edu/~odlyzko/doc/history.communications2.pdf

[19] CAIDA: Top problems of the Internet and What Sysadmins and Researchers Can Do to Help, 2003, http://www.caida.org/outreach/presentations/2003/netproblems_lisa03/netproblems_lisa2003.pdf

[20] Morin, M.: Peer-to-Peer File Sharing; Port Hopping and Challenges to Traffic Control Methodology, 2003, http://www.sandvine.com/solutions/pdfs/

[21] Wired News: Florida Dorms Lock Out P2P Users, 10/2003, http://www.wired.com/news/digiwood/0,1412,60613,00.html

[22] News.com: Stealth P2P Network Hides Inside Kazaa, 2002, http://news.com.com/2100-1023-873181.html

[23] Cisco IOS QoS, accessed 11/20/2003, http://www.cisco.com/warp/public/732/Tech/qos/

[24] www.ellacoya.com, accessed 11/20/2003

[25] News.com: P2P caching: Unsafe at any speed?, 7/2003, http://news.com.com/2100-1025_3-1027508.html

[26] www.joltid.com, accessed 11/21/2003

[27] Singh, S., Ramabhadran, S., Baboescu, F., Snoeren, A.: The Case for Service Provider Deployment of Super-Peers in Peer-to-Peer Networks, 2003, Workshop on Economics of Peer-to-Peer Systems

Peer to Peer and Content Distribution

Evgenia Daskalova
Research Scientist
Networking laboratory
Helsinki University of Technology
P.O.Box 3000, FIN 02015 HUT

Abstract

This paper considers the current problems of content distribution and the significant share of peer-to-peer traffic in today's Internet. The peer-to-peer technology is not new but has been widely popularized over the last few years because of the different useful applications that have been developed and successfully deployed in the Internet. An understanding of different content distribution systems such as HTTP web traffic, Content Distribution Networks and peer-to-peer file sharing is necessary for proper evaluation of the available technologies in this area. The main focus of the paper is to consider peer-to-peer from the point of view of content distribution systems.

1 Introduction

Peer-to-peer (P2P) systems in the Internet have raised a lot of interest recently. Although the popularity of this technology came with the distribution of audio files by Napster [15] that launched in 1999, there are other meaningful reasons for this technical phenomenon. Peer to peer is an alternative to the traditional client/server model that is suffering from some limitations as the Internet distributed environment is growing. In P2P systems every node acts both as a client and a server and that creates the possibility of high utilization of computing resources.

The peer-to-peer model is playing an important role in content distribution. The Internet has spread significantly over the last decade. There are much more users with high requirements and that brings many new challenges. One of them is the challenge of delivering increasingly complex data. This need has led to the development of clusters of thousands of nodes, global scale content distribution networks and more recently P2P file sharing structures. These content distribution systems are rapidly changing the nature of Internet content distribution and traffic characteristics. They bring new opportunities and solutions to many ascending technical problems.

This paper is organized as follows. Section 2 presents an overview of the different content distribution systems that are considered in other parts of this paper. Section 3 gives a traffic analysis regarding different key parameters of content distribution systems. In Section 4 the potential of P2P systems is studied by estimating business and technical advantages. Some implementations of peer-to-peer technology for the purpose of content distribution are also presented. Section 5 charts some future research directions on the P2P technology. Finally Section 6 concludes the paper.

2 Overview of content distribution

The Internet is full of different rich content that users would like to access. This content could be web pages that contain text, images, Java applets, frames and other objects, as well as MP3 files, audio presentations, video and stored virtual reality. The Internet architecture is such that every user could get whatever content he or she wants and wherever it is located if the user has appropriate access rights. But some time limitations need to be considered. The path that this content could travel to reach the user request could be a low-speed link that has large transmission delays; there could be also a congested link that causes long queuing delays and dropped packets. Another possibility is that the Web server that contains a wanted object is overloaded with requests, so the new request will suffer from long delays. In order to reduce these large delays, the strategy of replicating the content on one or more servers in the Internet has been used. Usually users are connecting to those servers that contain a copy of the wanted content and are located near by them, so that the server provides a shorter response time for the request.

Content distribution is a mechanism for (1) replicating content on multiple servers in the Internet and (2) providing requesting end systems a means to determine the servers that can deliver the content fastest [1]. The content distribution industry has started to expand after late 1990s and today we are witnessing a huge growth especially in the distribution of audio and video content.

Content distribution schemes could be classified into three main categories: Web caching; Content Distribution Networks (CDNs) and peer-to-peer file sharing. A brief introduction of all of them is presented in the next sections.

2.1 Web caching

A web cache is a network entity that satisfies HTTP requests on behalf of an original server. Users are configured in a way that their HTTP requests are directed to caches that are maintained usually by their Internet Service Provider (ISP). If the desired content is not there, the request is forwarded to the original server and retrieved from there, but also the proxy server saves a copy of the object to be used for future requests. Figure 1 shows the basic operation of web caching. The proxy server is located between the clients and the original web servers.

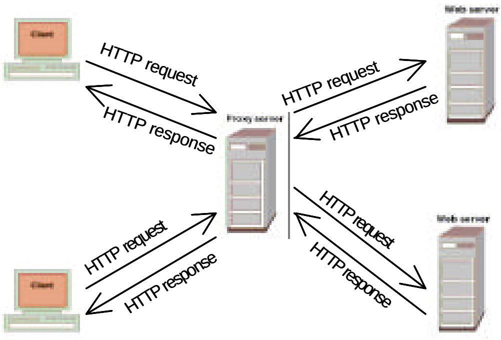


Figure 1: Client requesting through Web cache

Web caching is a form of content distribution because it replicates the content into the cache, near to the clients. Also the web cache is acting both as a server and a client. The major benefits of web caching are: reduced response time for user requests; reduced traffic in the Internet. One can claim that web caching is a good infrastructure model for distribution of content.

A lot of research has been done in this area, mainly focusing on different techniques of optimizing the performance and scalability and evaluating the cost benefits, but also for implementing cooperative caching or global caching structures. Various parameters that influence the efficiency of web caching such as cache-hit rate and cache size have been also analyzed [3,4,5].

Web caching is the first way to deal with performance and network resource utilization issues related to the growth and increased popularity of the Internet.

2.2 Content Distribution Networks

Content Distribution Network (CDN) is an architecture consisting of servers and other web based network elements that are arranged in a way for providing efficient delivery of web content. Although CDNs and web caching are similar, different business models are behind them. For a CDN company such as Akamai [6], the paying customer is the content provider such as Yahoo. A CDN company installs hundreds of servers in the Internet; afterwards it replicates its customers' content on them and finally provides mechanisms so that the user request for content will be satisfied with short delays [1]. CDNs are used not only for web content such as images but also for the distribution of streaming audio and video content that require additional support of streaming control protocols. Figure 2 gives an example of interaction between an original server in Europe and a CDN distribution node that replicates the content to CDN servers located in different continents around the world.

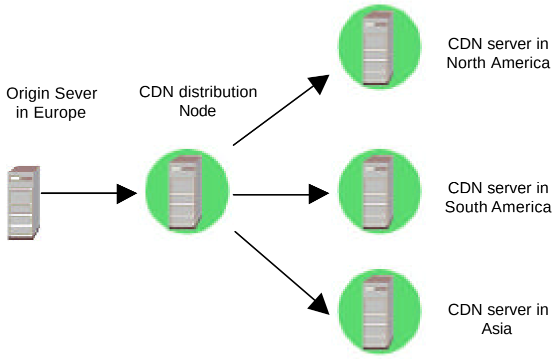


Figure 2: Interaction between the content provider and a CDN Company

Research in this area has been focusing on the effectiveness of this type of content distribution by measuring different performance characteristics [7]. Despite of some difficulties found in DNS redirection techniques that reflect in latency and a bottleneck in selecting the optimal server from the available ones, the advantages of CDNs for reducing the response time are reasonable for their usage and this has lead to their wide spread implementation over the last few years.

2.3 Peer to Peer File Sharing

Peer to peer file sharing is the third possibility for content distribution. The peers that are PCs at the edge of the network can retrieve objects directly from each other. Therefore, P2P takes advantage of the resources such as bandwidth, storage and CPU and allows a large amount of peers to distribute various content. This makes P2P file sharing a highly scalable technology. P2P systems offer an alternative to the traditional client/server model as every node could act both as a client and as a server, capable of running the file transfer protocol in both directions [8].

There are different important aspects regarding P2P file sharing such as communication and networking issues, security, privacy, anonymity and copyright infringement.

The easy part in P2P file sharing is how the content is transferred, but before that there is the difficult part of finding where the desired content is being located. There are three main architectures of content location that have been defined in [1].

2.3.1 Centralized directory

One of the first approaches is the centralized directory that became popular with Napster [15], which was the first successful case deploying a wide scale P2P application for MP3 distribution. In this design the peers contact a centralized directory server that is responsible for collecting information from the connected peers about their activeness and available content for sharing. The main drawbacks concerning this architecture are the single point of failure due to the possibility of the directory server to go down; performance bottleneck because of the huge amount of connected users that the database has to handle and the copyright infringement because of the lack of possibility of supervising the content that P2P users are exchanging and that caused Napster [15] to be shut down in 2002.

2.3.2 Decentralized directory

In the Decentralized directory design a certain number of peers are designated to be group leaders that maintain the database where the information of the active peers and their content is stored. The peers and their communication relationship form an abstract, logical network, called overlay network, which is evolving and highly dynamic. KaZaA [9] application took this approach and has become popular in 2001-2002. This architecture overcomes the disadvantages of the centralized directory shown above. However, there are still some drawbacks with this approach concerning the complexity of the protocols that are used, the group leaders that could become bottlenecks and also the presence of the bootstrapping node that works similarly to a server. The bootstrapping node responds with the IP address of one of the group leaders when a new peer wants to join the network. Peers use DNS to locate them. Therefore, bootstrapping nodes have to be always on in the network.

2.3.3 Query flooding

The third architecture is so called Query Flooding. This is a fully distributed approach for content distribution and Gnutella [10] is using it. The topology of the overlay network is flat and unstructured, every peer is equal and there are no group leaders. For object location the query-flooding mechanism is used. A limit of the radius for the query flooding is implemented to respond to the problem of scalability in Gnutella. This approach reduces the query traffic and the possibility of overloading the network. Therefore this simplified design has become highly attractive and widely adopted despite of the fact that the protocols used to maintain the overlay network are fairly complex. The Gnutella design also requires some bootstrapping nodes, so that a new peer can establish an initial connection with an existing peer in the network. Building a P2P application without any bootstrapping node is a challenging problem.

All of the above briefly presented architectures have been deployed in Internet and were used a lot, despite of some of their limitations.

3 Traffic Analysis of Content Distribution Systems

Traffic analyzing of different content distribution systems is introducing a view for understanding how these systems impact the Internet. It also provides many specific insights related to the network provisioning, bandwidth utilization, performance management, traffic balance, connectivity complexity and vulnerability, QoS and other important network aspects.

Although a lot of research has been performed in the area of measurements and traffic analysis concerning different network elements and applications, including P2P file sharing, content distribution systems have been analyzed separately rather than being compared.

3.1 Bandwidth Comparison

Bandwidth is one of the most important parameters of traffic analysis. Research done in this area proved that nowadays the Internet traffic is consisting of a huge part of P2P traffic and future trends are that this part is increasing.

Traffic could be classified, based on application criteria, into two major categories such as HTTP traffic and non-HTTP traffic. The HTTP traffic could be subdivided into WWW, Akamai content distribution network [6] and P2P file sharing. For such application based classification one can use port numbers from the TCP header. Therefore, more concrete definition could be the following:

- **WWW:** HTTP traffic on port 80, 8080 or 443 which is not served by any CDN.
- **Akamai:** HTTP traffic on port 80, 8080 or 443 that is served by an Akamai server
- **P2P:** HTTP traffic on ports 6346, 6347 or 1214 (etc.) that is traffic generated by Gnutella or Kazaa (etc).

A good comparison of the three main content distribution systems defined as above in [2] in terms of bandwidth showed that P2P systems generate a large percentage of bytes exchanged in both directions over a one week period as can be seen in Figure 3.

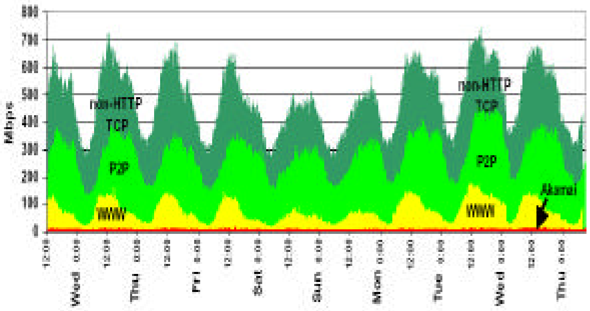


Figure 3: Bandwidth comparison of the HTTP and non-HTTP traffic

It can be observed that the smallest bandwidth consumer is Akamai, which constitutes around 0.2% of the HTTP traffic, followed by WWW with 14,3% and P2P traffic with 43%. The rest of the traffic, around 43%, is non-HTTP traffic such as streaming, news, mail and others. There are no significant changes in consumed bandwidth during the different days of the week in this traffic trace. These results of bandwidth consumption reveal substantial changes in content distribution systems usage in the Internet by indicating a huge amount of P2P traffic. Therefore, it could be concluded that the P2P traffic has become the largest contributor of HTTP traffic.

3.2 Distributed objects overview

The users demand for retrieving different types of content of bigger and more bandwidth consuming objects has been increasing over the last few years. A difference between content distribution systems can be observed by comparing them in terms of size and types of objects that are being distributed.

3.2.1 Size of objects

As one can see in Figure 4, the size of the objects retrieved by P2P systems such as Kazaa and Gnutella are three orders of magnitude bigger than the objects retrieved by WWW and CDN systems [2]. There is also an obvious difference in the proportion of the percentage of the small objects as the size increases. The median object size of these P2P systems is approximately 5MB. This indicates that the potential growth of such systems will influence the overall Internet traffic performance.

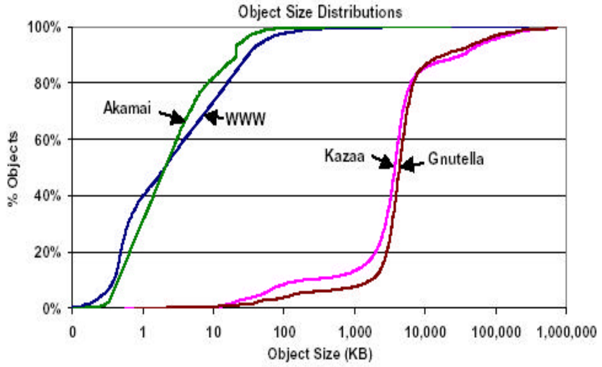


Figure 4: Object size distribution

3.2.1 Type of objects

The objects that are being distributed in Internet could be considered in terms of type. Figure 5 shows a comparison between content distribution systems and types of objects generally classified into text, images, video, audio and others. WWW traffic is composed mainly of text while Akamai traffic is composed of images. P2P traffic is composed of video and audio content that are the heaviest bandwidth consumers in the Internet.

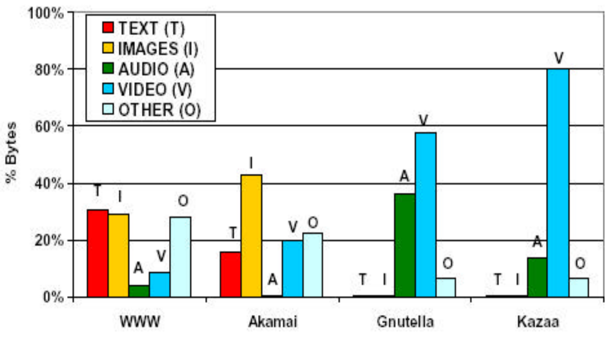


Figure 5: Downloaded bytes by objects distribution

3.3 Clients comparison

Another important aspect of traffic analysis is the distribution of number of clients and bandwidth consumption. This dependence is tightly connected also with the type and size of distributed objects that were presented and analyzed above.

In both WWW and Akamai systems the number of clients is slowly increasing with bandwidth consumption. P2P systems are having a lesser number of clients that are generating a huge amount of traffic according to [2].

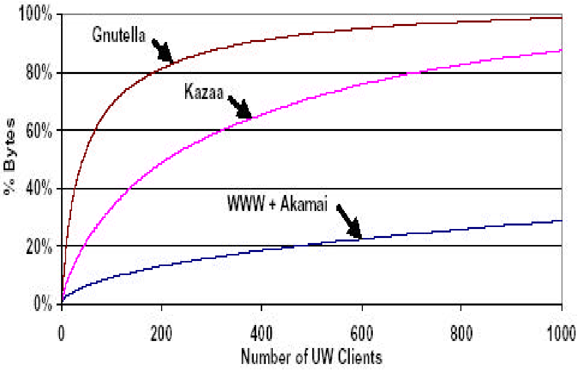


Figure 6: Downloaded bytes by clients distribution

Figure 6 exhibits this dependency. It is obvious that for the same number of top bandwidth consuming UW (University of Washington) clients, the downloaded bytes percentage is dissimilar for the different content distribution systems. For example in the case of WWW and Akamai, the top 200 clients account for around 13% of WWW and Akamai traffic; for Kazaa the top 200 clients account for 50% of KaZaA traffic. In both Kazaa and Gnutella P2P systems, a small number of clients account for a large portion of traffic. Therefore, it is easy to conclude that P2P clients have a greater impact on HTTP traffic nowadays than other types of clients.

3.4 Connections duration

The duration of the connections is also an interesting parameter that has to be considered in traffic analysis. A method to measure the connection duration would be to count the number of HTTP flows for the different content distribution systems. We could analyze the relation by observing Figure 7. The chart gives the number of concurrent HTTP transactions that are active at the same time for the different content distribution systems [2].

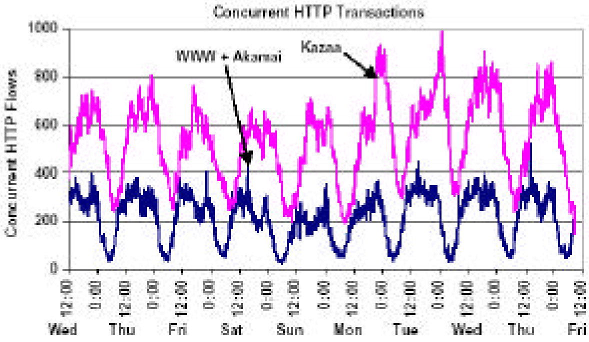


Figure 7: Concurrent HTTP transactions

The numbers of concurrent P2P HTTP flows, represented by the Kazaa application in this example, are two times higher than those generated by WWW and Akamai. This dependency is also valid during the whole week with slightly different variations. It could be concluded that even a single P2P system, like Kazaa, generates more HTTP flows, thus longer connections, than all the other content distribution systems combined. If future studies show that this trend is stable, we would expect that the further utilization of P2P systems could significantly affect the Internet infrastructure, as it could create possibilities for further optimization of routing based on the increased number of flows.

These traffic analyses have implications for the Internet as a whole but also for exploring in particular the scalability of the P2P systems. Kazaa and Gnutella have been considered as examples of some of the most popular and successful P2P applications. As a content distribution system, Akamai, that is one of the baggiest and excellent companies, was investigated. The traffic traces were captured from a real network monitoring between University of Washington (UW) and the rest of the Internet. The results are useful as a background for proper evaluation of the importance of all content distribution systems and good understanding of the research done so far in this area. Some future directions could also be pointed out based on these traffic analyses and the need for more measurements for better and deeper performance evaluation on content distribution systems.

4 Functional analysis of content distribution systems

This section is focused mainly on the Content Distribution Networks and peer-to-peer file sharing. Both systems provide challenging techniques and algorithms for optimizing the performance of content distribution in the Internet. Therefore, their potential role in the future have to be analyzed by considering technical advantages that could be utilized and used by researchers. Some proposal solutions that are available already are also briefly presented. A particular emphasis is given on how these technologies could work together.

4.1 Technical benefits of CDN

Content Distribution Networks were developed originally for the purpose of saving money; conserving bandwidth and improving the Internet traffic performance. CDNs are placed at the edge of the ISP (Internet Service Provider) network. That reduces the distance between the user and content provider and increases the speed of delivery. The advantages of using CDNs are both for the content providers and end users. New CDN types are trying to face the problem of supporting delivery of any digital content such as rich media and large files. By implementing secure and software based distributed networking techniques it is possible to assure distribution of on demand streaming with a high quality of service and reduced bandwidth cost [6, 16].

There are two fundamental elements that describe the value of CDN according to [12]:

- **Scale.** This is the outsourcing infrastructure. CDN allows multiple surrogates (servers holding copies of content) to act on behalf of the original server.
- **Reach.** This is the property of improved content delivery. CDNs are placed near to the end user, overcoming network size, network congestion and network failures.

Besides the technical benefits of using CDNs, there are some limitations that characterize their usage. Operating a CDN is not an easy task for content providers because CDN are relatively complex systems and have to connect points that could be far away geographically.

4.2 P2P potential

P2P is not only important because of the traffic characteristics, some of which were presented in Section 3 and their influence on the Internet. There are other different value and technical aspects that support the potential of P2P. They help to understand why P2P technology is popular nowadays and how many, new, powerful and advanced P2P solutions for various applications could be created.

P2P potential to meet many business and personal needs is generating interest in this technology. The Internet today is full of digital information that is replacing the traditional media such as paper. P2P offers the information and services that are most important to users. Another key determinant of value is cost. It has many dimensions such as time, required skills, standards and accessibility. P2P is aimed to use peers own personal computer resources. Therefore, the cost of maintaining web sites that include connectivity, programming, operating system maintenance and hardware cost is reduced. P2P technology saves time, one of the critical factors nowadays, as it enables users to connect directly to the information that they need, eliminating the delays. P2P gives control over own valuable information, as it resides on users personal computers rather than on public web servers. Variety is another key value issue. P2P offers the full richness of Internet to be enabled and utilized by its users [11].

On the other side is the technical value of P2P that is important for understanding the potential of this technology and the future directions. Some major points are shortly introduced below.

- **P2P leverages Internet openness.** P2P is free to deploy any type of interaction formats and protocols on the Internet. This freedom could produce waves of innovation.
- **P2P enables technology standards.** Standards offer a solution to the openness of Internet networking. The development of standards has progressed rapidly and is becoming a big enabler of P2P technology[3].
- **P2P leverages personal computer hardware.** Following the Moore's law the personal computer capabilities and performance have improved. Personal computers are efficiently multitasking, storing huge amounts of data and handling high-speed network communications that P2P systems could use.
- **P2P leverages personal computer information and application services.** This is the ability of users to create useful information or powerful services on their own personal computers. So, these P2P participants will offer potential for promising P2P future.
- **P2P gives high search performance.** The search algorithms used in P2P are showing a high performance for finding the desired information by users.
- **P2P offers a fully distributed symmetric architecture.** P2P escapes from the traditional client/server model. The architecture influences the overall services and reliability performance.

These are some of the technical aspects of P2P, summarized on a very general level. They form the basic premises for success and future potential of peer-to-peer systems considered in details in [11].

4.3 Using P2P technology for Content Distribution Internetworking

Content Distribution Internetworking (CDI) is a model purposed by IETF, given in [12], for the ambition of achieving scalability and effectiveness in user response time by cooperation of multiple CDNs. This interest in investigating interconnecting of content networks is for offering better overall service to the users and better performance of the networks.

CDNs and P2P systems differ from each other by many aspects, such as traffic aspects some of which were considered in Section 3, but also from implementation and design point of view. Despite of these differences there are some similarities given below that could be used for the idea of cooperation of these systems [13].

- **Request routing:** This is a task of selecting the most desired content that is satisfying a given

[3] Editors note: Rather one could observe that the P2P model is a perfect platform for any company that creates its own P2P application to establish its own protocol standards provided that standards exist for content formats.

user request. Could be implemented in a centralized or decentralized manner. It is performed in both P2P and CDN systems.

- **Data delivery:** Both P2P and CDN systems in general aim to deliver different types of data. It could be located on one server (as with centralized architecture) or on multiple replica servers.
- **Content replication:** Content replication is implemented for optimizing user access to requested data. Both P2P and CDN systems use this approach. Resources could be classified as static and dynamic that could be maintainede by both systems.

Those common characteristics of P2P and CDNs could be used as a basis for investigating the possibility of using P2P techniques in the design of the request routing, data delivery and replication mechanisms in CDI.

4.4 Cooperative Networking overview

Cooperative Networking (CoopNet) is an approach, proposed by the Microsoft Research center, to content distribution that combines aspects of infrastructure-based (Content Distribution Networks) and peer-to-peer content distribution [14]. CoopNet compliments the traditional client-server communication rather than replaces it. It is focusing on the distribution of streaming media content, both live and on demand. The type of content that is possible to be located on one server creates a problem when there is a high volume of requests from its clients. By using CoopNet where end hosts cooperate with each other, the network performance perceived by all is improved.

4.4.1 Flash crowd problem

The CoopNet model is proposed for the use of solving the flash crowd problem at web sites. A flash crowd refers to a rapid and dynamic increase of the volume of requests to the server due to some events of great interest that are not planned. As a result, the server is overwhelmed and its response time is increasing. The CoopNet approach is addressing this flash crowd problem by using the clients that have already downloaded the content, to serve the new clients that are not able to access the original server. Peer cooperation is only invoked during the duration of the flash crowd event.

4.4.2 Operation of CoopNet

CoopNet is focusing on reducing the bandwidth demands on the server, as this was figured out to be the major bottleneck when a flash crowd problem occurs in the network. During the flash crowd, the server redirects some or all requests of the clients to the other clients that have the desired URL already downloaded.

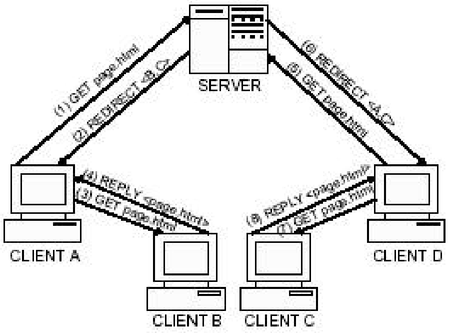


Figure 8: The basic operation of CoopNet

Figure 8 illustrates the basic operation model of CoopNet. The numbers in parenthesis indicate the order of the steps that have to be performed in the process of redirection. The Figure also shows the type of messages that are exchanged between clients (peers) and the server. The clients send a modified request header to inform the server about their willingness to join CoopNet. The server saves the IP addresses of some of those CoopNet clients, which have requested files recently. Afterwards, the server uses, some randomly selected IP addresses in the redirection message. It is quite likely that at least some of these peers will be able to serve a new request. [14].

An interesting matter is the peer selection problem. This is a question of how a peer that receives a redirection message from a server will decide which peer to contact in order to download the desired content. The scheme proposed by [19] and used by the researchers, which work on this model, is to employ a so-called multi-pronged approach. According to that model, a peer that has a request for content is connecting to the topologically close other peers that participate in CoopNet.

4.4.3 Practical evaluation of CoopNet

A prototype implementing CoopNet has been built in the Microsoft research center for the purpose of performance evaluation. One interesting analysis was done based on traces gathered from the MSNBC website on September 11, 2001 flash crowd in New York City.

Finding content

Some results considering efficiency of content finding are given in Figure 9. There are two parameters defined which show how often content can be retrieved from a peer group, rather then from the original loaded server. One is the new content hit rate that is the fraction of requests for new files that could be served by hosts in the peer group. Another is the fresher content hit rate which is the fraction of time that a fresher copy of time could be found in the peer group. These two parameters have to be high in order to conclude that CoopNet is an efficient mechanism.

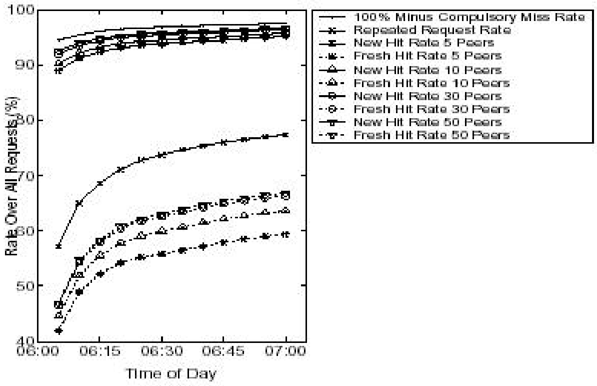


Figure 9: Average hit rates observed

Different scenarios have been created and analyzed as can be seen in Figure 9. The upper bounds of lines are representing the so-called optimistic scenario, when the files are not modified between accesses. The down bounds of lines are the results from the pessimistic scenario, when a file is being updated during requests on the server each second. As a result of the analysis done in [14] by taking into account different scenarios and number of peers willing to cooperate, it becomes clear that cooperation among a small group of peers with CoopNet is the most effective case.

Load on peers

Another interesting parameter that was analyzed is the load of the peers. This is an important factor for the purpose of maintaining high performance because the CoopNet peers are contributing their resources to the system. Figure 10 shows the result obtained by [14] from the experimental evaluation of CoopNet.

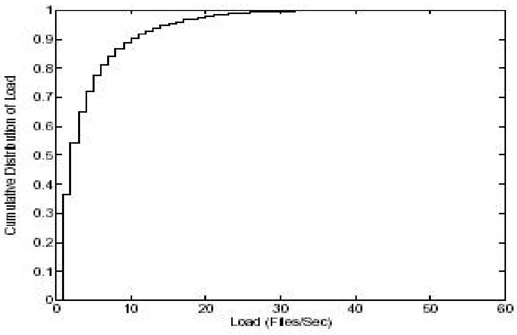


Figure 10: Load of peers during busy periods

The results show the network bandwidth overhead. Peers are idle and do not serve content 80% of the time, but during the remaining 20% they are highly loaded. This could lead to a flash crowd problem at peers. Therefore, we need to investigate further the load distribution and peak bandwidth requirements for peers that participate in CoopNet.

These parameters along with the metric of finding the nearby peer for cooperation and duration of time of activity of a peer influence the efficiency of this model. In summary, although there are still some limitations of this model, it could be a practical solution for the problem of flash crowd in the Internet in future.

5 Future directions

There are many future challenges for the peer-to-peer technology. It is very difficult to predict, even impossible, in which direction P2P systems will take on a quick pace. Research in this area, so far, has shown that the technology behind P2P is giving a lot of opportunities. Some of them have already been taken, and users are exploring and benefiting from them, others are waiting to be discovered.

One important question is whether P2P is going to overtake the content distribution market? Peer to peer is a type of content distribution system; therefore we believe that in the future this particular application will be more researched. New models have to be built that can take advantage of the potential of P2P. Some of them could be in cooperation with other existing content distribution methods. One possible direction is to use P2P technology in order to design and implement a model for content delivery for educational purposes, such as for the universities. That could be a way to provide students, professors and researchers with the information that they need to find or exchange. And the benefits for them will be time, cost savings, high search performance and improved effectiveness from an end-to-end user P2P solution.

Another question is how much the size and growth rate of the content distribution systems will satisfy different users requirements and needs? The demand for multimedia content and larger files is growing rapidly also for the purpose of gratifying the dynamically changing entertainment markets. The optimisation assumptions for the Internet architecture may need to be modified from the original concept of carrying small and short files. Therefore the distribution problem that will satisfy different market trends needs to be researched further. It is necessary also to consider how much customers are willing to pay for the desired content if they can get it legally and conveniently with low latency and avoiding network congestion.

6 Conclusion

This paper examined peer-to-peer file sharing from the point of view of available content distribution systems in the Internet today. A theoretical background was briefly presented in the second Section for the purpose of understanding these systems architectures.

Traffic analyses were used for evaluating how peer-to-peer systems perform compared with Content Distribution Networks such as Akamai and web caching in respect to different parameters. It becomes obvious that P2P systems present a significant part of the current Internet traffic. Therefore, more measurements and traffic analysis need to be done in the future for better understanding of different content distribution systems.

The potential that P2P technology offers to the users and researchers have been considered as well as those for CDNs. This paper showed that there are other directions for File Sharing P2P, like Content Distribution Internetworking and Cooperative Networking, besides just distributing illegal audio and video files. These applications combine methodologies and provide solutions for overtaking some limitations and problems in the Internet distribution systems.

In summary, we considered that peer to peer technology offers many traffic and functional challenges and improvements to the existing content distribution systems. Therefore we believe that ISPs should conduct further research of P2P traffic to create and maintain a proper understanding of the various traffic parameters that could be used to optimize their networks. We also believe the content providers should invest in P2P systems, as they prove to be more efficient technology for sharing information, and could minimize their operational costs also. We truly believe that as the technology itself is a logical evolution in the existing ways to share information, if the image of P2P systems is cleared of the negative association with piracy and illegal software, this could be one of the leading and most exciting technologies for further research and use in the Internet.

List of acronyms

CDI: Content Distribution Internetworking
CDN: Content Distribution Network
CoopNet: Cooperative Networking
CPU: Central Processing Unit
DNS: Domain Name Server
HTTP: Hypertext Transferred Protocol
QoS: Quality of Service
IETF: Internet Engineering Task Force
ISP: Internet Service Provider
P2P: Peer-to-Peer
TCP: Transmission Control Protocol
URL: Uniform Resource Locator

References

[1] J. Kurose, K. Ross, Computer Networking, A Top-Down Approach Featuring the Internet, second edition, Addison Wesley, 2003

[2] K. Aberer, M. Hauswirth, An Overview on Peer-to-Peer Information Systems, Swiss Federal Institute of Technology, EPFL, 2002

[3] S. Gadde, J. Chase, M. Rabinovich, Web caching and content distribution: A view from the Interior, 2000

[4] R. Lancellotti, M. Colajanni, B. Ciciani, "A Scalable Architecture for Cooperative Web Caching", Proc. of Workshop in Web Engineering, Networking 2002, Pisa, May 2002

[5] M. Bradley, Duska, D. Marwood, M. J. Feeley, The Measured Access Characteristics of the World-Wide-Web Client Proxy Caches, Proceedings of the 1997 Usenix Symposium on Internet Technologies and Systems (USITS-97)

[6] Akamai. http://www.akamai.com/

[7] K. L. Johnson, J. F. Carr, M. S. Day, and M. Frans Kaashoek, The measured performance of content distribution networks, Computer Communications, 24(2), 2001.

[8] K. Aberer, M. Hauswirth, An Overview on Peer to Peer Information Systems, Swiss Federal Institute of Technology (EPFL), Switzerland, 2002

[9] Kazaa. http://www.kazaa.com/

[10] Gnutella. http://www.gnutella.com/

[11] D. Moor, J. Hebeler, Peer-to-peer: Building Secure, Scalable and Manageable Networks, McGraw-Hill, 2002, ISBN 0-07-219284-4

[12] M. Day, B. Gain, G. Tomlinson, P. Rzewski, A Model for Content Internetworking (CDI), IETF, RFC 3466, Feb.2003

[13] E. Turrini, F. Panzieri, Using P2P Techniques for Content Distribution Internetworking, Proceedings of the Second International Conference on Peer to peer computing, 2002

[14] V. Padmananbhan, K. Sripanidkulchai, The case for Cooperative Networking, Peer-to-Peer Systems: First International Workshop, IPTPS 2002

[15] Napster. http://www.napster.com

[16] CenterSpan. http://www.centerspan.com/

[17] http://www.peertopeersource.com/

[18] V. Padmananbhan, K. Sripanidkulchai P. Chou, Distributing Streaming Media Content Using Cooperative Networking, Technical report MSR-TR-2002-37, Microsoft Research, 2002

[19] B. Krishnamurthy, J. Wang, On Network Aware Clustering of Web Clients, ACM SIGCOMM, August 2001

Building up Trust Collaboration in P2P Systems based on Trusted Computing Platform

Zheng Yan
Nokia Research Center
zheng.z.yan@nokia.com

Abstract

Peer-to-peer computing has emerged as a significant paradigm for providing distributed services, in particular collaboration for content sharing and distributed computing. But this kind of computing paradigm suffers from several drawbacks as well that obstructs its wide adoption. Lack of trust between peers is one of the most serious issues, which causes security challenges in the P2P systems. This paper studies the feasibility to build up trust collaboration based on trusted computing platform (TCP) in peer-to-peer systems. Based on the analysis, the author believes that the TCP technology is a promising solution that can overcome many P2P security challenges.

1 Introduction

Peer-to-peer computing has emerged as a significant paradigm for providing distributed services, in particular collaboration for content sharing and distributed computing. Generally, a P2P system consists of a decentralized and self-organizing network of autonomous devices that interact as peers. Each peer acts as both client and server to share its resources with other peers.

There is no widely accepted definition of peer-to-peer computing or networking. In [1], P2P is described as an environment where computers connect to each other in a distributed environment that does not use a centralized control point to route or connect data traffic. In [2], the author argues that it is healthy and desirable not to be locked down by a rigid definition because this computing model is rapidly evolving.

There are many variants of applications that employ P2P technologies. Typically, those applications fall into two categories: content sharing and distributed computing. P2P permits direct sharing of documents, multimedia and other files between network peers. Napster, Gnutella and Freenet are examples of P2P content sharing applications. NetBatch and SETI@home are examples of P2P-based distributed computing. P2P allows the use of the resources of idle hosts to conduct computing tasks.

Peer-to-peer computing has significant benefits including scalability, low cost, robustness and ability to provide site autonomy. With the great success of many P2P applications, it becomes more and more popular, even towards mobility. However, this approach also suffers from several drawbacks that influence its wide adoption. Security, interoperability, bandwidth and resource search are main challenges that retard its wide usage [3].

The paper is organized as follows. In section 2, the author studies the trust problems, which cause many security issues in the peer-to-peer networking. Based on the introduction of trust computing technology in section 3, the author proposes trust collaboration architecture based on TCP in Section 4. The architecture is analyzed in section 5. Section 6 discusses some related work, followed by conclusions provided in the last section

2 Problem Statement

Recent studies discover many problems in the P2P systems [3, 4]. One of the major ones is Security. Normally, the P2P applications give computers or devices access to other machines' resources, e.g. hard drives, which can be vulnerable to attacks.

There are a number of reasons why security is crucial in the P2P systems. We summarize the reasons as follows.

Firstly, downloading files from other machines makes the systems vulnerable to viruses. For example, Gnutella users experienced VBS_GNUTELWORM virus [3].

Secondly, it is important that communicating computers or devices have the ability of authenticating the identity of each other when they engage in collaboration.

Thirdly, the availability of resources is seriously threathened by DoS attacks by overloading some nodes. A chosen-victim attack in Gnutella is a specific example. Generally, it is easy to defend against external attacks, but difficult to fight attacks raised from internal malicious nodes in P2P systems.

Fourthly, when online users become more concerned about privacy, some of them may hesitate to use the P2P services. They will not accept a technology if personal information will be exposed without any control. A more secure P2P infrastructure is expected.

Finally, intellectual property management and digital rights management (DRM) are highly required in P2P systems. We have to restrict access to shared contents according to copyrights and legal usage rights. Flexible DRM control is a necessity in the P2P systems.

A peer-to-peer network is a self-organizing system. Such a system lacks trust among peers since sharing resources and access must be granted to unknown peers. The whole P2P network environment is made up of heterogeneous hardware and software components with dynamic capability (e.g. bandwidth). The peers could come and leave the connection randomly. In addition, the scale of the network could be in millions or as few as containing two peers. Most possibly, peers holding different local policies are moving at separated locations.

Fundamentally, sharing and making use of resources requires collaboration among peers in the P2P systems. The key of above security problems is to build up trust collaboration in the P2P systems.

3 Trusted Computing Platform

The current technologies for trusted computing platform are quite similar [5, 6]. The typical TCP technologies are specified in the specifications of TCG (Trusted Computing Group) [7]. TCG aims to enhance the overall security, privacy and trustworthiness of a variety of computing devices.

TCG's Trusted Computing Platform (TCP) builds its promise of a trusted platform on the basis of some hardware – the Trusted Platform Module (TPM). In short, TPM is the hardware that controls the boot-up process. Every time the computer is reset, the TPM steps in, checks TPM and then verifies the BIOS before letting boot-up continue. The BIOS is assumed to verify the operating system, the operating system is assumed to verify every bit of software that it can find in the computer, and so on.

The TPM chip (separate from the processor) and other TCP modules simply allow all the hardware and software components to check whether they have woken up in the trusted states. If not, they should refuse to work. It also provides secure storage for confidential information. Figure 1 illustrates the basic structure of TCP and the relationships between its compo nents.

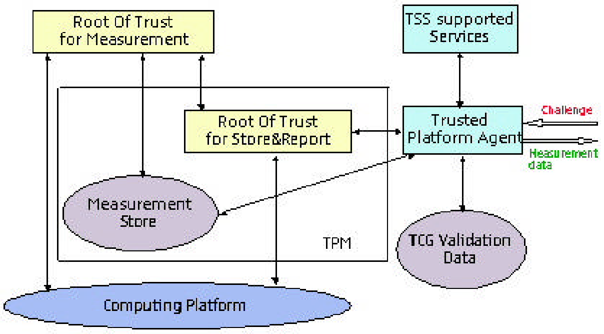


Figure 1: Basic structure of TCP and components relationship

Simply speaking, there are four basic functions provided by TCP.

3.1 Authenticated booting

An authenticated boot service monitors what operating system software was booted and gives applications a sure way to identify which OS is running. This is achieved by keeping audit logs of the boot process.

When booting up, a TPM chip takes charge. The chip checks it sees the boot ROM it expects, executes it, and measures the state of the machine; then it checks the first part of the operating system, loads and executes it, and checks the state of the machine; and so forth. That is, the BIOS boot block checks the hardware specification of the PC against a known safe integrity metric; and should that match, the system then authenticates the user. It then checks the operating system loading software. The OS loader, once proven safe, checks the OS kernel. The kernel knows how to check the list of legitimate software, which in turn can use OS resources to authenticate local and remote data.

The TCP hardware keeps a tamper-evident log of the boot process, using a cryptographic hash function to detect any tampering with the log. The above 'check' is conducted like this: the loader calculates the hash code of the next SW contributor logging it in the TP measurement store. If the value derived from the log is the same as that reported by TPM, the check is passed.

This is helpful for the TC hardware to know what software configuration is running on a machine. What is more, the TCP hardware can make the configuration known to others. This is done through certificating digitally the configuration. Two levels of certifying are provided in TCP.

Certifying OS configuration
On this level the system uses a private key only known by TPM to sign a certificate that contains the configuration information, together with a random challenge value provided by the challenger.

The configuration can be presented to any challenger (user, a program running on another computer). The challenger (provided that it generated the random challenge) can verify that the certificate is valid and up-to-date, so it can know what the machine's configuration is.

Certifying applications
In many cases, there is a stronger desire to certify the presence and configuration of application programs. Application configurations are certified through a two-layer process. TPM certifies that a known OS version is running and then the OS can certify the application's precise configuration.

3.2 Encryption services

Encryption service is the second major offer of TCP. It allows data to be encrypted in such a way that it can be decrypted only by a certain machine, and only if that machine is in a certain configuration.

This service is implemented by a combination of hardware and software facilities. The TPM hardware maintains a 'master secret key' for each machine, and it uses the master secret to generate a unique secret encryption key for every possible configuration of that machine. Thus, data encrypted for a particular configuration cannot be decrypted when the machine is in a different configuration.

This service can be extended from OS level to applications. This ensures that encrypted data can only be decrypted by desired version of desired applications when running on top of desired OS and on desired machine. So, we can transmit data to a remote machine in such a way that that data can be decrypted only if the remote machine is in a certain configuration. An application can also control the data in a similar way when encrypting data before writing to disk, so that the data can be decrypted only by the same version of the same application, running on the same machine.

All in all, encryption service provides a special control on digital data through encryption to make it accessible only when an expected platform environment is present.

3.3 Privacy support

The TCG specification provides a method for obtaining an anonymous user identity certificate from a privacy CA over a secure channel. The procedure is described in the following.

The TPM sends the public key (of the user that desires a certificate) and three credentials to a privacy CA. The three credentials include:

- *A public key certificate*: it is the endorsement certificate issued by the entity that endorsed or certified the TPM. This is most possibly issued by the device manufacturer. It contains a null subject and the TPM public endorsement identity's public key, among other things.

- *The first attribute certificate*: the platform credential containing a pointer to the endorsement certificate that uniquely identifies the platform's endorser and the model – hardware and software versions, TPM details, platform compliance with the TCG specifications, etc.

- *The second attribute certificate*: the conformance credential, that asserts that the named TPM complies with the TCG specification.

The CA receives these three certificates, and verifies the information. Then the CA creates a TPM identity credential and sends it to the client via the secure channel. The TPM identity credential contains a null subject and the public key sent by the user in the certificate request.

This procedure ensures that anonymous certificates are only issued to compliant devices.

3.4 DRM support

The TCG specifications present several problems regarding to DRM and competition as well as open source GNU public license (GPL). A TCG-enabled OS could prevent the user from running "unapproved" applications. Through extending the encryption service offered by TCP, the TCG-enabled computing platform could control digital contents access, execution and use of programs as well as the operation of the system according to the specified rules.

4 Building up Trust Collaboration on TCP

With TCP compatible devices in the P2P system, it will be easy to build up trust collaboration to support secure P2P applications. In what follows, the author proposes a P2P infrastructure based on TCP and analyses how this infrastructure can solve the security problems listed in section 2, therefore support trust collaboration in P2P systems.

4.1 Definitions

Due to multiplicity of meanings associated with the word 'trust' and its derivatives, it is essential to establish certain set of definitions that can be used throughout the paper.

Definition 1: Trust

The working definition of trust used in this paper is the confidence of an entity on another entity based on the expectation that the other entity will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other entity.

Definition 2: Trust modeling

Trust modeling is a technical approach to represent trust for digital processing. A trust model specifies, evaluates and sets up trust relationships among entities.

Definition 3: Trusted computing platform

To be a trusted computing platform, a computing system must behave the way it is expected to behave for the intended purpose. The TCG's TCP technology ensures this through a set of hardware and software mechanisms for authenticated booting, platform integrity attestation and providing encryption and decryption attached to platform specific configurations.

Definition 4: Trust collaboration

Herein, trust collaboration is defined as interaction, communication and cooperation conducted according to the expectations of involved entities. For example, the shared contents in P2P systems should be performed (consumed and used) following the content originator's expectation without violating any copyrights. In peer-to-peer systems, the trust collaboration requires autonomous control over network resources by any one peer at any remote peer if needed.

4.2 Trust modeling

Based on the methodology presented in [8, 9], a trust model is proposed for the P2P system according to its specific characteristics: lack of trust among peers, dynamic topology, and heterogeneous peer devices with different local policies. As shown in Figure 2, each peer device is independently located inside a personal trusted bubble: the basic unit that represents a peer. In the bubble, the owner of the peer device illogically fully trusts the device, which is responsible for the communication with other peers and network resource organization. Among bubbles, logical and rational trust relationships should be attested.

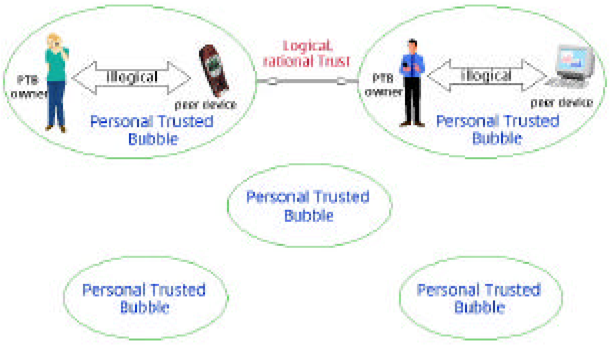


Figure 2: Peer-to-peer system trust model

4.3 Trusted computing infrastructure for P2P

Based on the above trust model, the author proposes a trusted computing infrastructure for the P2P system. In this infrastructure, each peer device is TCP compatible and has an internal architecture as shown in Figure 3.

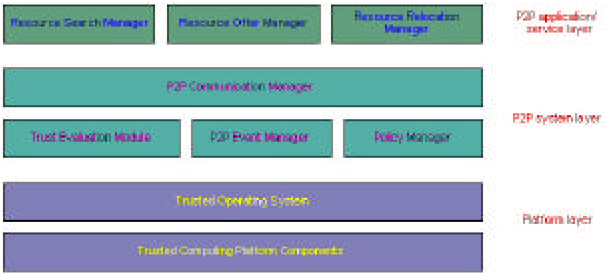


Figure 3: Architecture of P2P peer device built on TCP

There are three layers in this architecture:

Platform layer contains TCP components as shown in Figure 1 and operating system that is booted and executed in a trusted status attested and ensured by the TCP components.

P2P system layer contains common components required for trusted P2P communications. Those components are installed over the platform layer and ensured running in the trusted status through the TCP components and OS. Communication manager is responsible for various P2P-related communications, while the trust evaluation module is applied to evaluate trust relationship with any other peer before any security related decision is made [9-11]. The trust evaluation model should cooperate with policy manager and event manager in order to work out a proper trust evaluation result.

P2P application/service layer contains components for P2P services. Taking resource sharing as an example, this layer should contain components like resource search manager, resource offer manager and resource relocation manager. Like the system layer, all the components in this layer are attested by the platform layer as trusted and executed as expected. Any malicious change could be detected by the platform layer and rejected.

4.4 Trust collaboration

The trust collaboration in the proposed P2P system infrastructure can be supported as follows:

- Each peer device can verify that another peer device is working in its expected trust status.

  Building up on the TCP technology, each peer device with the underlying architecture can ensure every P2P component on the device is working in a trusted status. It can also challenge any other device and attest that it is working in its expected status.

- Each peer can manage the trust relationship with other peers and therefore it can make the best decision on security issues in order to reduce the potential risks.

  Based on the trust evaluation mechanism [9-11] embedded in the trust evaluation module, each peer can anticipate potential risk and make the best decision on any security related issues in the P2P communications. In addition, the trust evaluation is conducted in the expected trust environment, thus the evaluation results could be trusted.

- Resources are offered under expected policies.

  This includes two aspects. One is that the resources are provided based on copyright restrictions. Those contents that cannot be shared should not be disclosed to other peers. The other is that the resources are provided with some limitations defined by the provider. The encryption services offered by the TCP can cooperate with the resource offer manager to offer protected resources and ensure copyrights and usage rights.

- Resources are relocated safely and consumed as the provider expects.

  The trust attestation mechanism offered by the TCP can support the resource relocation manager to attest that the downloaded contents are not malicious code through trust platform challenge and code integrity hash verification. In addition, the resources are used in expected ways that are decided according to either copyrights or pre-defined usage restrictions by the providers. This is ensured ahead of consuming by the encryption mechanism offered by the TCP.

- Personal information of each peer should be controlled according to expectations.

  The resource offer manager in the proposed architecture can cooperate with the TCP components. It encapsulates the personal information based on the policies offered by the policy manager and only trusted resource search manager can access it. The trusted resource search manager is the expected P2P application component that can process the encapsulated personal information according to the processing requirements pre-defined by the personal information owner.

With the strong support of the TCP components, any P2P components can and only can execute as expected and process resources in the expected way. What is more, with the trusted platform support on the trust evaluation, the peers could communicate in the most trusted way.

5 Further Discussion on Security Challenges

In this section, the author discusses how the TCP based P2P infrastructure could help overcome security challenges presented in section 2.

- Virus vulnerability

  In the proposed architecture, platform integrity challenge and attestation could ensure that any virus does not affect the underlying communicating platform. In addition, any downloaded file from the resource relocation manager should be further attested by the TCP components to ensure that the code is safe. The hash code of expected data is used to conduct the verification. The TCP technology can ensure that the virus challenge can be processed accordin to expectation.

- Identity authentication

  TCP components provide secure storage to save a unique platform ID and also provide support to assign various aliases on this ID for privacy purposes. If every peer device is TCP compatible, they can authenticate each other based on the platform ID and its alias.

- The risk raised by malicious peers could be greatly reduced based on trust evaluation. Due to the importance of the trust evaluation, it requires sound protection to ensure its correct process. The TCP components in the proposed architecture provide a good running environment and ensure this environment for trusted trust evaluation.

- One important mechanism that can be supported by the TCP based P2P architecture is privacy. A different alias of the platform ID can be used for different purposes. The alias could be also attached to some specified platform configurations or application configurations to support restricted P2P services. In addition, the encryption services can also be applied into the user profile (that stores the user information) in order to control in which kind of situation, the information inside the profile can be accessed.

- DRM is strongly supported in the TCP based P2P architecture through encryption service mechanism. Most importantly, this mechanism can be further extended to attach encryption to specified usage rights and specified content consuming software to ensure the expected processing environment of the shared contents.

6 Related Work

There is some related work conducted in the literature.

In [1], an open-source framework JXTA was proposed to support programming secure peer-to-peer applications. It contains a set of protocols to realize secure peer-to-peer connection. It also supports certificates provided by peers, which behave as the internal CA. This programming platform is based on the Java technology, which is a pure software solution on P2P security. It lacks support on DRM, virus control and private data Spam.

In [10], the MOTION architecture was proposed to realize access control over mobile P2P environment. But it has no support on autonomous access control over already shared resources.

In [2], collaboration is thought of as humans involved in the P2P systems interacting with each other in a near real-time manner. The concept of collaboration is different from what we defined in Section 4. In this paper, we pay more attention to the collaboration that can be conducted automatically among P2P devices. Two collaboration frameworks were introduced in [2]: Endeavors and Avaki. Both frameworks and the Proem architecture introduced in [11] build upon a software platform and use a software solution to control access. This kind of framework cannot fully support access control on remote resources that have been shared automatically during network collaboration.

In [12], a hybrid architecture mixing a trusted centralized control with untrusted peer-to-peer components was proposed for an enterprise P2P scenario. In this architecture, distributed resource usage is adaptive to the trustworthiness of the distributed components. The central control component is in charge of coordinating the interaction with the external services and the untrusted peer-to-peer components. In this model, the overall architecture is adaptive to trust and reliability assessment. Trust of an untrusted component is assessed through evidence collection. But this paper did not discuss how to support trusted trust assessment, which is considered in our paper.

There is some work on building up a new trust model for the P2P systems. In [13], a trust model based on trust-based group (*troups*) is suggested. This model supports transitive trust in its author’s opinion. But this model needs special protocol to support dynamic membership inside the troups. Compared to our trust model, this model is more complicated to manage. According to [14], trust is not always transitive. Therefore this model needs further study in order to prove the transitivity property.

A line of trust modeling work for P2P systems is based on reputation [15-17], in which reputation is the main factor that is deployed for trust evaluation among peers or domains. This kind of P2P trust modeling is similar to ours. But the trustworthiness in this kind of P2P system is based on trust evaluation, not on a trusted computing platform.

In [18], a protocol for anonymous trust management was proposed. It provides mutual anonymity for both trust host (that manages the trust ratings of the P2P peers) and trust querying peer in order to secure trust management in P2P distributed systems. Our proposal is different from this solution in that each peer is supposed to run independently and anonymously if needed and our proposal is supported by uniformed platform architecture, not a protocol.

7 Conclusions

TCP technologies are under-development in the industry and academy in order to provide more secure and better trust support for future digital devices, such as PC, mobile phone, and PDA, etc. TCP tries to solve existing security problems by hardware trust. Although it is still in its infancy and may be vulnerable to some hardware attacks [22], it has advantages over many software-based solutions.

In this paper, the author introduced a perspective of building up trust collaboration in a P2P system based on trusted computing platform. Through a uniformed TCP compatible P2P device architecture, many security challenges can be overcome, therefore, realizing trust collaboration in this kind of trust lack network environment. In addition, the TCP based P2P system can also support network self-organization and automatic network resource management as well as privacy if needed. It has potential advantages over other solutions, especially when the TCG standard work is done and many industry digital device vendors (such as Microsoft, IBM, HP, Intel, etc.) offer TCP-compatible hardware and software in the future.

8 References

[1] Yeager, W., Williams, J.: Secure peer-to-peer networking: the JXTA example, IT Professional, Volume: 4 Issue: 2, March-April 2002 Page(s): 53–57.

[2] Barkai, D.: Technologies for sharing and collaborating on the Net, Proceedings of First International Conference on Peer-to-Peer Computing, 2001, 27-29 Aug. 2001 Page(s): 13–28.

[3] D. Clark, Face-to-Face with Peer-to-Peer Networking, Computer, Vol. 34, No.1, January 2001, pp.18-21 12.

[4] Daswani Neil, Garcia-Molina Hector and Yang Beverly: Open Problems in Data-Sharing in Peer-to-Peer Systems, In ICDT, 2003.

[5] Edward W. Felten, Understanding Trusted Computing – Will it Benefits Outweigh Its Drawbacks, IEEE Security & Privacy, May/June 2003.

[6] England Paul, Lampson Butler, Manferdelli John, Peinado Marcus, Willman Bryan: A Trusted Open Platform. IEEE Computer Scciety, p55-62, July 2003.

[7] Trusted Computing Group (TCG) main specification, version 1.1a, Nov. 2001. http://www.trustedcomputinggroup.org/

[8] Yan Z. and Cofta P.: Methodology to Bridge Different Domains of Trust in Mobile Communications, The First International Conference on Trust Management, Greece, 05. 2003.

[9] Yan Z., Zhang P., and Virtanen T.: Trust Evaluation Based Security Solution in Ad Hoc Networks, The Seventh Nordic Workshop on Secure IT Systems, NordSec 2003, Gjovik, Norway, 10, 2003.

[10] Fenkam, P., Dustdar, S., Kirda, E., Reif, G., Gall, H.: Towards an access control system for mobile peer-to-peer collaborative environments, Enabling Technologies: Infrastructure for Collaborative Enterprises, 2002. WET ICE 2002. Proceedings. Eleventh IEEE International Workshops on, 10-12 June 2002 Page(s): 95–100.

[11] Gerd Kortuem, Jay Schneider, Dustin Preuitt, Thaddeus G. C. Thompson, Stephen Fickas, Zary Segall: When Peer-to-Peer comes Face-to-Face: Collaborative Peer-to-Peer Computing in Mobile Ad hoc Networks, The First International Conference on Peer-to-Peer Computing (P2P'01), Sweden, August 2001.

[12] Mont, M.C., Tomasi, L.: A distributed service, adaptive to trust assessment, based on peer-to-peer e-records replication and storage, Proceedings of The Eighth IEEE Workshop on Future Trends of Distributed Computing Systems, FTDCS 2001, 31 Oct.-2 Nov. 2001, Page(s): 89–95.

[13] Gokhale, S., Dasgupta, P.: Distributed authentication for peer-to-peer networks, Proceedings of Symposium on Applications and the Internet Workshops, Jan. 2003, Page(s): 347–353.

[14] Cahill, V., Gray, E., Seigneur, J.-M., Jensen, C.D., Yong Chen, Shand, B., Dimmock, N., Twigg, A., Bacon, J., English, C., Wagealla, W., Terzis, S., Nixon, P., Di Marzo Serugendo, G., Bryce, C., Carbone, M., Krukow, K., Nielson, M.: Using trust for secure collaboration in uncertain environments, Pervasive Computing, IEEE , Volume: 2 Issue: 3 , July-Sept. 2003, Page(s): 52–61.

[15] Li Xiong, Ling Liu: A reputation-based trust model for peer-to-peer e-commerce communities, IEEE International Conference on E-Commerce, CEC 2003, 24-27 June 2003, Page(s): 275–284.

[16] Yao Wang, Vassileva, J.: Trust and reputation model in peer-to-peer networks, In Proceedings of Third International Conference on Peer-to-Peer Computing, 2003. (P2P 2003), 1-3 Sept 2003, Page(s): 150–158.

[17] Azzedin, F., Maheswaran, M.: Trust modeling for peer-to-peer based computing systems, Proceedings of Parallel and Distributed Processing Symposium, 2003, April 22-26, 2003, Page(s): 99–108.

[18] Singh, A., Ling Liu: TrustMe: anonymous management of trust relationships in decentralized P2P systems, in Proceedings of Third International

Conference on Peer-to-Peer Computing, 2003. (P2P 2003). Sept 2003, Page(s): 142 –149.

[19] Crichton, C., Davies, J., Woodcock, J.: When to trust mobile objects: access control in the Jini^{TM} Software System, Proceedings of Technology of Object-Oriented Languages and Systems, 1999. TOOLS 30. , 1-5 Aug. 1999, Page(s): 116 –125.

[20] Druschel, P., Rowstron, A.: PAST: a large-scale, persistent peer-to-peer storage utility, Proceedings of the Eighth Workshop on Hot Topics in Operating Systems, 20-22 May 2001, Page(s): 75 –80.

[21] Caronni, G., Waldvogel, M.: Establishing trust in distributed storage providers, Proceedings of Third International Conference on Peer-to-Peer Computing, 2003. (P2P 2003), 1-3 Sept 2003, Page(s): 128 –133.

[22] Andrew “Bunnie” Huang, The Trusted OC: Skin-Deep Security, Computer (Oct. 2002) vol.35, no.10, p.103-5.

Open Problems in Peer-to-Peer Systems: Quality of Service and Scalability

Jani Lakkakorpi
Nokia Research Center
P.O. Box 407
FIN-00045 NOKIA GROUP
Finland
jani.lakkakorpi@iki.fi

Abstract

The volume of peer-to-peer (P2P) traffic is already now a major issue in many access networks – mostly due to large file sizes. Thus, P2P traffic needs to be addressed somehow. Some Internet Service Providers (ISPs) have already come up with various techniques that alleviate this problem. We shall evaluate the different approaches and view the technical solutions needed. Another open, peer-to-peer system related, problem is scalability – mostly in content search. In this area, research is (and has been) particularly active. In this paper, we compare a couple of different approaches on bringing scalability to Gnutella-like file sharing systems.

1 Introduction

A topic like "Open Problems in Peer-to-Peer Systems" leaves quite a lot of freedom for the author. However, in this particular case, those open problems that are covered by the other papers in this seminar report [1] – such as legal, security or economics related issues – are excluded from this paper. In this paper, we will concentrate on two problems that we find most interesting: Quality of Service (QoS) – or the lack of it – and scalability (in content search) issues in peer-to-peer networking.

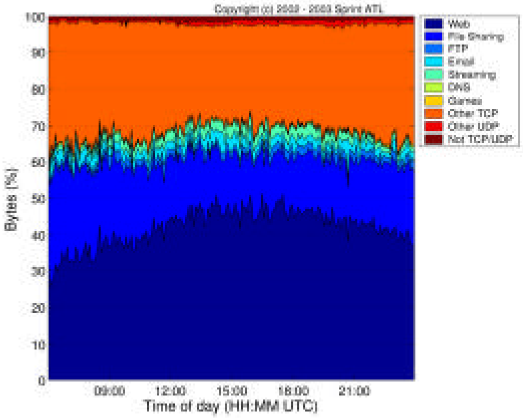


Figure 1: Sprint, NYC, April 7th, 2003, 2.5 Gbps link: traffic (bytes) breakdown by application [3]

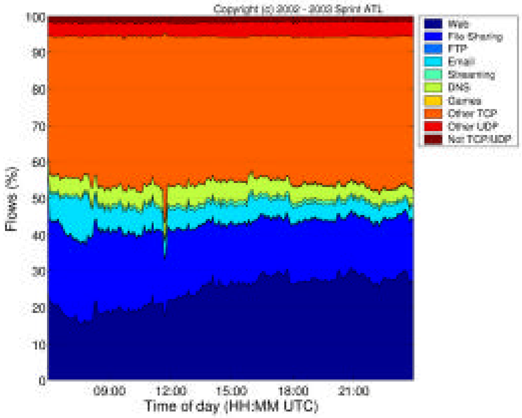


Figure 2: Sprint, NYC, April 7th, 2003, 2.5 Gbps link: traffic (flows) breakdown by application [3]

The share of peer-to-peer (P2P) traffic of overall network traffic (almost anywhere) is growing rapidly; see e.g., [2] and [3]. These statistics are also cited in a peer-to-peer related tutorial [4], where it is noted that over the last three years peer-to-peer services have become one of the most import sources of Internet traffic. Already now, peer-to-peer file sharing protocols cause close to 20% of the total traffic[4] volume at some US core routers (see Figures 1 and 2 and Table 1).

[4] P2P traffic is not asymmetric like web traffic, which makes the problem even worse. The introduction of asymmetric broadband access, however, does not seem to be a good remedy as it leads into growing traffic volumes from "outside" as the desired content is not that easily available locally [5].

Table 1: Sprint, NYC, April 7th, 2003, 2.5 Gbps link: traffic breakdown by application [3]

Category	Packets (%)	Bytes (%)	Flows (%)
Web	39.14	42.94	25.10
File Sharing	**18.13**	**17.43**	**17.86**
FTP	1.11	2.05	0.29
Email	5.27	2.82	5.27
Streaming	2.52	2.43	0.94
DNS	1.52	0.44	4.42
Games	1.29	0.27	0.14
Other TCP	25.60	29.41	40.47
Other UDP	3.23	1.13	3.93
Not TCP/UDP	2.18	1.07	1.58

Even more dramatic figures (from access networks) are presented in [6], where Mellin claims that peer-to-peer applications constitute 60% of the total traffic volume in the case of a typical “euro-ISP”. This is seriously threatening the position of web traffic as the number one traffic source in the Internet. Of course, this is also threatening the timely delivery of web pages and other interactive content – if all traffic is treated equally in traditional Best Effort fashion and/or if the link capacities are not upgraded.

A main characteristic of peer-to-peer services is their highly distributed, serverless architecture using autonomous peers. Instead of using the traditional client/server paradigm, peer-to-peer services are decentralized; they typically form application-specific virtual network structures – overlays. It is possible for the peers to join or leave the overlay any time. Naturally, the attractiveness of a peer-to-peer service increases with the number of peers contributing to the service. However, this is also the case with the amount of signaling traffic needed. Thus, scalability is one of the major problems in peer-to-peer networking.

This paper is organized as follows: section 2 deals with the QoS related problems introduced by peer-to-peer networking, section 3 concentrates on the scalability problems e.g., in content search, while section 4 concludes the paper with discussion.

2 Quality of Service – for All Users

As noted in the introduction, the impact of high traffic loads generated by different peer-to-peer applications is already present; network operators are complaining about high traffic volumes and users about decreasing reliability and attractiveness. Thus, it is necessary to implement an efficient – and preferably simple – performance and traffic management system.

From the users’ perspective, P2P performance management should provide a consistent quality of locating and accessing resources as well as minimize the negative effects of P2P on other network services. From the operators’ perspective, traffic management should lead into an optimal traffic distribution and efficient network operation [4]. However, we believe that these two goals are too much contradictory. In this section, we shall try to answer the following question: “How the operators can provide consistent Quality of Service for all users?”

2.1 Different Attempts to Control the Volume of Peer-to-Peer Traffic

Reference [7] illustrates the problem of growing peer-to-peer traffic and its control through simple examples like “people who hog up lanes on the information highway are about to run into some roadblocks”. In more technical terms, this means that some Internet Service Providers are already preparing a series of measures to control the amount of bandwidth people can use. One of these ISPs, Cox Communications Inc. [8], will implement a tiered pricing system, where heavy bandwidth users shall be charged more than normal users. The “premium version” will cost $80 to $90 per month, while the “low-cost choice” is $25 to $30 per month.

An alternative (or complementing) solution – that some ISPs are also considering – would be to implement a system that tells the ISP who is a bandwidth-hogging peer-to-peer user and who is just a harmless Web surfer. Different users would be controlled in different ways. Reference [7] mentions a company called P-Cube Inc. [9] that has a product (called Engage) that is able to slow down certain downloads by heavy users so that they do not disturb the Web surfing of other users. The product is able to differentiate between regular download or browsing sessions and peer-to-peer downloads without actually identifying the content being downloaded. It is also possible to limit the total P2P traffic to a certain percentage of link capacity.

It is believed (at least by one Gartner Group analyst) that the different tiered pricing schemes – possibly with multiple tiers – will be implemented before the technically more advanced schemes such as P-Cube's technology [7]. However, we believe in Differentiated Services.

2.2 Is DiffServ the Answer to Peer-to-Peer Bandwidth Hogging?

Simply “throwing bandwidth” is not a particularly good solution, since the increased bandwidth use by peer-to-peer services is driven by larger media files, which leads to an endless cycle.

2.2.1 Different Solutions

In our opinion, Quality of Service – and more specifically Differentiated Services (DiffServ) [10] is the best solution for the peer-to-peer bandwidth hogging problem.

DiffServ has already been around for a while – at least in the IETF standards. However, the implementation phase has not taken off properly. Several reasons for this, e.g., the potential complexity [11] of DiffServ, have been presented.

Now there finally appears to be a strong motivator to start implementing DiffServ – at least on the worst bottleneck links in access networks[5]. It is not necessary to apply DiffServ on such links that have enough bandwidth during the busy hour.

What would have to be done is simply to "downgrade" the treatment of such flows that utilize too much network resources (and not only P2P flows as in P-Cube's technology). In practice, this could be done with a single Assured Forwarding (AF) [12] class – AF1, for example. In this case, the default Per-Hop Behavior (PHB), Best Effort (BE), would be treated like AF11[6], which means that in the case of congestion, packets marked as AF13 and AF12 would be dropped before packets marked as AF11 or BE. This can be implemented, e.g., by using Weighted Random Early Detection (WRED), which means having own Random Early Detection (RED) [13] process for each drop precedence level within an AF class.

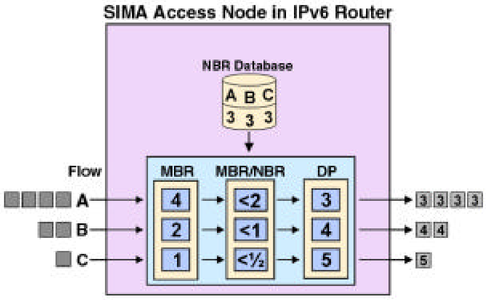


Figure 3: SIMA access (edge) node [15]

Naturally, the proposed scheme would require flow measurements and packet marking at the network edge. Moreover, different customers could buy different volumes of bandwidth. In this case, the measured bit rate at the network edge would be compared to the purchased capacity and possible packet (re-)marking would occur according to that ratio. This is essentially what Simple Integrated Media Access (SIMA) [14] is about[7]. Figure 3 and Figure 4 illustrate the edge and core functionalities of SIMA. MBR stands for Measured Bit Rate, NBR for Nominal Bit Rate (the purchased bandwidth) and DP for Drop Precedence. The core node functionality in SIMA is slightly similar to WRED – the difference being that there is no queue size averaging in SIMA. Moreover, the dropping process is not AF compliant as both RT and NRT buffer occupancies affect the dropping decision of any packet.

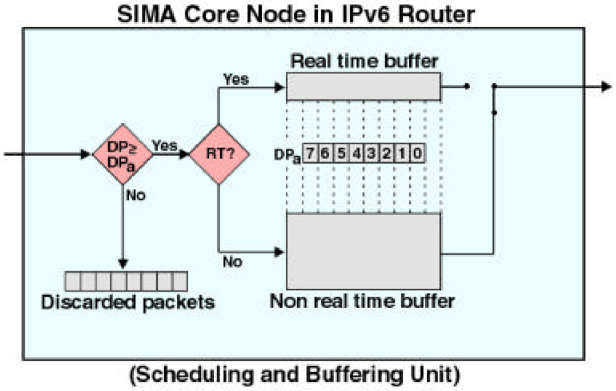


Figure 4: SIMA core node [15]

2.2.2 Simulation Experiment

In order to verify our claims, we conducted a simple simulation experiment that compared the performance of SIMA to Best Effort – with the presence of web and "P2P" traffic. In the SIMA case, all users were "donated" a NBR of 50 kbps, while in the Best Effort case there was neither packet marking nor differentiated packet treatment. We constructed a four-node topology (see Figure 5) with a 10 Mbps / 250 ms bottleneck link between the middle nodes[8] and put 80 web surfers (with a realistic web traffic model) and 20 "P2P" users, each downloading a file with "infinite" size (using FTP), to node 0. All content was downloaded[9] from node 2. Simulation duration was 300 seconds. All traffic sources were launched at a random time during the first 10 seconds. As can be seen from Figure 6, the bottleneck link is fully occupied all the time with both schemes.

We emphasize that the following results (see Table 2) can be viewed as trend-setting ones only. The reason is that only single (and relatively short) simulation run was executed. TCP goodput (i.e. what the application gets) for web surfers is probably the best indicator of experienced Quality of Service. We can see that there is a clear difference in favor of SIMA. The Best Effort case seems to favor bandwidth-hogging "P2P" users. This happens most probably because the FTP file downloads are "infinite" in our case, whereas web surfers stop to read the pages before downloading new ones – short page downloads cannot properly compete with long-lasting FTP downloads. The introduction of SIMA (with equal Nominal Bit Rates) completely changes the situation: a NBR of 50 kbps slows down the FTP flows and favors the HTTP flows – due to dynamic drop precedence calculation and differentiated packet dropping. However, more studies are needed to verify these results.

[5] In core links, there should not be any problems.

[6] In a DiffServ router, it is possible to provide the same treatment for packets with different DiffServ Code Points (DSCP).

[7] SIMA also allows real time vs. non-real time traffic separation but that is not particularly important in this context.

[8] The other two links: 100 Mbps / 50 ms.

[9] Thus, our "P2P" users were essentially "free-riders".

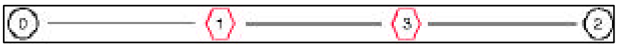


Figure 5: Simulation topology

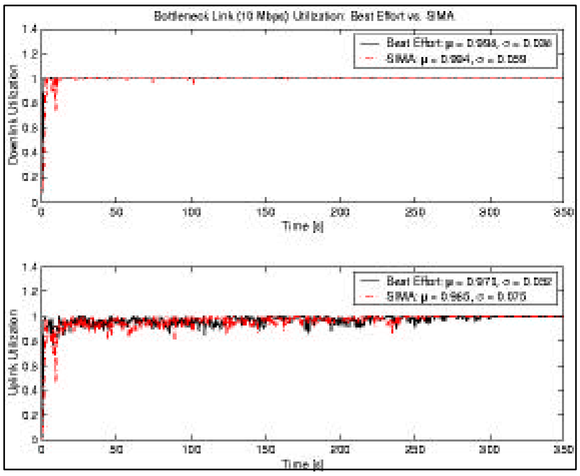


Figure 6: Bottleneck utilization

Table2: Simulation results: SIMA vs. Best Effort

	Best Effort	SIMA

Goodput, HTTP	μ = 55.5 kbps,
	σ = 46.9 kbps
		μ = 325.9 kbps,
		σ = 292.9 kbps

Goodput, FTP	μ = 475.5 kbps,
	σ = 35.8 kbps
		μ = 468.5 kbps,
		σ = 13.6 kbps

Recv. bytes, HTTP	12.5 M	12.9 M
Recv. bytes, FTP	412.5 M	412.0 M
Packet loss, HTTP	20.8%	14.2%
Packet loss, FTP	5.7%	7.4%

Traffic shaping at the network edge (which is most probably just what the ISPs offering tiered pricing schemes are doing) is not a good remedy for peer-to-peer traffic, since that would artificially limit the maximum bit rates – also when there is plenty of room on the bottleneck link. Traffic shaping would suit better for admission controlled traffic, e.g., for video streaming that would be mapped to a slightly overprovisioned AF class.

Once DiffServ has been introduced because of peer-to-peer bandwidth hogging, the operators and ISPs could start to offer more QoS sensitive services, e.g., Voice over IP (VoIP), too. In traditional Best Effort networks the strict delay, jitter and packet loss requirements of VoIP would be very hard to guarantee. With Expedited Forwarding (EF) [16] and admission control[10] that could be done as well.

3 Scalability in Content Search

The previous section discussed the Quality of Service issues with peer-to-peer traffic. However, the solutions that were proposed are in fact general and they apply to any bandwidth-hogging flows. It just happens to be the case that at the moment peer-to-peer traffic is growing so rapidly that it is the one we need to tame first.

This section is definitely more peer-to-peer oriented as it deals with the content search scalability issues in peer-to-peer networking. As noted in the introduction, in a peer-to-peer system, autonomous computers pool their resources (e.g., files, storage and compute cycles) in order to inexpensively handle tasks that would normally require large servers. However, the scale of these systems and the lack of centralized control cause difficult performance challenges [17].

3.1 A Brief Overview on Gnutella

Since Gnutella [18] is a very popular P2P application and it suffers from scalability problems, a short overview on Gnutella is given before going any further.

The purpose of Gnutella is distributed and anonymous file sharing by exploiting the unused storage on edge nodes [19]. These servents (SERVer + cliENT) operate without any central control. Naturally, in this kind of circumstances node discovery has to be performed before any content downloading is possible. In the node discovery phase, PING/PONG messages are sent with TTL (Time To Live) to limit the broadcasting range. A short time memory of already seen messages prevents re-broadcasting.

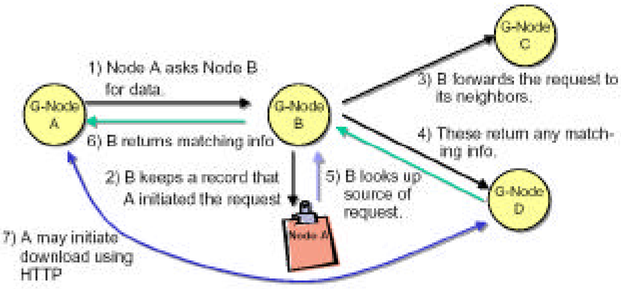

G-Node A
G-Node B
G-Node C
G-Node D
Node A
1) Node A asks Node B for data.
6) B returns matching info
2) B keeps a record that A initiated the request
3) B forwards the request to its neighbors.
4) These return any matching info.
5) B looks up source of request.
7) A may initiate download using HTTP

Figure 7: Search query and download phases in Gnutella [19]

The search query phase is illustrated in Figure 7. After a record that matches the request has been found, it can be downloaded using HTTP.

The scalability problem here is evident: e.g., with a TTL of 10 and assuming that every node broadcasts to six other nodes, we get 6^{10} messages. Moreover, the TTL cannot be too low either as this would mean low search horizon. The following subsections try to tackle this (or slightly similar) problem.

It is worth noting here that the search in Gnutella is essentially different from Napster (the pioneer of P2P file sharing), which utilized a centralized search based on file lists provided by each peer.

[10] Overprovisioning might be needed outside the operator's access network.

3.2 Attempts to Make Gnutella-like Systems Scalable

The number of results returned is an important QoS metric. However, in Gnutella-like systems, where there is high autonomy, we have a clear tradeoff between number of results and cost. Directed BFS (Breadth-First Traversal) technique in [20] attempts to minimize cost by sending messages to nodes with large collections only. In an another approach, concept-clustering networks [21], peers are clustered together according to interest, e.g., music genre, and queries are sent to the cluster that best matches the area of interest. Both of these techniques improve the tradeoff between cost and number of results, but are clearly not optimal. Performance of the directed BFS depends on the ad-hoc topology and is therefore unpredictable, while concept-clustering assumes that queries and interests happen to fall into single categories [17].

Many researchers, e.g., Stoica *et al.* [22], have proposed Distributed Hash Table (DHT) solutions to the wide-area file search problem. Contrary to that trend, Chawathe *et al.* want to preserve Gnutella's simplicity while proposing new mechanisms that will improve its scalability [23].

3.2.1 Distributed Hash Tables

Distributed Hash Tables have one basic operation, `lookup(key)`, which returns the value associated with the key. In peer-to-peer systems, the keys could be filenames and the values could be IP addresses of the nodes that store the associated files. This functionality allows building of Internet-scale facilities above DHTs, e.g., distributed file sharing.

The biggest driver for DHTs has been to make the Gnutella-like file sharing systems scalable. The research for DHTs has been very active during the past few years. Most DHT-related proposals use structured overlay networks where both the data placement and overlay topology are tightly controlled.

Chawathe *et al.* [23] note that the lookup operation DHTs use typically requires only $O(log\ n)$ steps, whereas Gnutella requires $O(n)$ steps to reliably locate a specific file. However, Gnutella-like designs are more robust (when taking into account that the median up-time for a node is only about 60 minutes [24]) and they support general search facilities – which are both important properties in P2P file sharing. It is true that Gnutella-like designs are worse than DHTs at finding "needles" (rare files) but this may not matter, after all, since most P2P queries are for "hay" (popular files). Thus, it is assumed that for mass-market file-sharing applications, it is more important to improve the scalability of unstructured P2P systems than switch to DHT-based systems.

3.2.2 Gia Approach

Partly building on prior research (e.g., [25], a preliminary proposal for incorporating capacity awareness into Gnutella), Chawathe *et al.* propose several modifications to Gnutella design that dynamically adapt the overlay topology and the search algorithms in order to accommodate the natural heterogeneity (in processing power, disk latency, access bandwidth etc.) present in most peer-to-peer systems. Chawathe *et al.* believe that the supernode[11] approach used, e.g., in KaZaA [26] is a step in the right direction for building scalable peer-to-peer file sharing systems. In [23], a new peer-to-peer file-sharing system, called Gia, is presented. Like Gnutella and KaZaA, Gia is decentralized and unstructured.

Chawathe *et al.* point out that in Gnutella-like systems, nodes quickly become overloaded when they are faced with a high query rate. Naturally, the problem gets worse as the size of the system increases. The first goal in designing Gia was to create a Gnutella-like peer-to-peer system that can handle high aggregate query rates. The second goal was to make Gia function well with increasing system sizes. To achieve the scalability, Gia avoids overloading any of the nodes by taking into account their capacity constraints.

As explained before (see Figure 7), Gnutella uses a flooding-based search in order to locate files within the peer-to-peer network. To locate a file, a node sends a query to each of its neighbors, which in turn send the query to their neighbors until the query reaches all clients within a certain distance (TTL) from the node that sent the original query. The described approach can find even the most rare files. However, the scaling problems are obvious. To alleviate this problem, Lv *et al.* [27] have proposed to replace flooding with random walks.

Random walks are a technique in which a query message is forwarded to a randomly chosen neighbor (instead of flooding the message to all neighbors) at each step until sufficient responses to the query are found. Although random walks result in better utilization of the peer-to-peer network than flooding, they have some associated problems. The first problem is that a random walk is essentially a "blind search" – at each step the query is forwarded to a random node without worrying how likely it is that the node will have responses for the query. Secondly, if a random walk query arrives at a node overloaded with traffic, the query may get queued for some time.

Said that, an ideal search protocol should somehow bias its random walks towards high-degree nodes. If the

[11] Designated "supernodes" have higher bandwidth connectivity. Pointers to each peer's data are stored on an associated supernode, and all queries are routed to supernodes.

neighboring nodes are arranged to be aware of each other's content, the high-degree nodes will most probably have pointers to a large number of files. Thus, the high-degree nodes will be more likely to have an answer matching any query. However, by favoring the high-degree nodes we ignore the problem of overloaded nodes. Actually, it could make things worse if the high-degree nodes do not have the capacity to handle a large number of queries.

The design of Gia takes into account the capacity constraints associated with each node in the peer-to-peer network and the node heterogeneity (processing power, disk latency, access bandwidth etc.) is exploited to achieve better scaling. The four key components of Gia design are the following:

1. **A dynamic topology adaptation protocol** ensuring that high capacity nodes are the ones with high degree (i.e. they are well connected) and that low capacity nodes are always close to higher capacity nodes. This should guarantee that the well-connected nodes, which will receive the most queries, have the capacity to handle them.
2.
3. **An active flow control scheme** in order to avoid overloaded hot-spot nodes. The flow control protocol adapts to heterogeneity by assigning flow-control tokens to nodes based on available capacity. Each Gia client assigns tokens periodically to its neighbors. A single token represents a single query that the client is able to accept.

4. **One-hop replication** of pointers to content. All nodes keep pointers to the content offered by their immediate neighbors. Since the topology adaptation guarantees that high capacity nodes are well connected (i.e. they have high degree), the one-hop replication scheme makes sure that high capacity nodes are capable of providing answers to more queries than low capacity nodes.

5. **A search protocol based on biased random walks** that points queries towards high-capacity nodes, as it is likely that they have the answers to most queries.

The proposed design has been tested through simulations and the results show three to five orders of magnitude improvement (compared to TTL-scoped flooding, random walks and "supernode" mechanisms) in total system capacity.

4 Conclusions and Discussion

It is important that the network operators are able to apply different methods, e.g., DiffServ in order to limit bandwidth hogging – independent of application type (P2P vs. other applications). SIMA, for example, might be an ideal solution – taking into account the results of our simulation experiment[12].

Moreover, it is equally important that the P2P protocols (especially the content search mechanisms) will scale. In our opinion, however, the argument presented in [23], "most P2P users search for common files", is a bit vague. For a file-sharing user, scalability of a protocol can hardly substitute for getting the desired results – even if they are "needles" i.e. rare files. Thus, there should be room for more DHT related research (see e.g., [22]) as well.

Standardization on P2P is not particularly active at the moment. Nevertheless, there is a Peer-to-Peer Research Group within the IRTF [28], which could mean (but not necessarily) that standardization within the IETF might start in the near future. Without going into details we simply note that the first proposal for research tracks in P2Prg [29] includes the following items:

- Overview of current P2P systems
- P2P Overlay Infrastructure
- P2P Mobility Device Requirements
- Meta-Data Strategies
- Namespaces: managed versus unmanaged
- Routing and routing primitives
- Peer Discovery/Resource Location/Presence
- Security issues

5 References

[1] R. Kantola *et al.*, "S-38.030 Postgraduate Course on Networking Technology: Peer-to-Peer Networking and Spam in the Internet", Fall 2003. `http://www.netlab.hut.fi/opetus/s38030/F03/`

[2] Information available at `http://netflow.internet2.edu/`

[3] Information available at `http://ipmon.sprintlabs.com/`

[4] K. Tutschku, "Network Efficient P2P-Services and their Management", *Tutorial in 18th International Teletraffic Congress (ITC) – Providing QoS in Heterogenous Environments*, Berlin, Germany, August–September 2003. `http://www3.informatik.uni-wuerzburg.de/staff/tutschku/ContentITCTutorial.pdf`

[5] P. Helenius, "Peer-to-Peer", *Laajakaista kotona – mitä laajakaistaan?*, Espoo, Finland, November 20, 2003. `http://akseli.tekes.fi/Resource.phx/tivi/nets/laajakaista.htx`

[12] However, we admit that more research is needed to verify the promising initial results.

[6] J. Mellin, "Vertaisverkot ja operaattorit", *Laajakaista kotona – mitä laajakaistaan?*, Espoo, Finland, November 20, 2003. http://akseli.tekes.fi/Resource.phx/tivi/nets/laajakaista.htx

[7] R. Cheng, "In the Pipeline: ISPs Poised To Slow Bandwidth Hogs", Dow Jones News wires, November 11, 2003. Available (for example) at http://www.p-cube.com/doc_root/news/In_News/WSJ.com%20-%2011.12.03.pdf

[8] Cox Communications Inc. http://www.cox.com/

[9] P-Cube Inc. http://www.p-cube.com/

[10] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An Architecture for Differentiated Services", Request For Comments 2475, December 1998. http://www.ietf.org/rfc/rfc2475.txt?number=2475

[11] G. Bell, "Failure to Thrive: QoS and the Culture of Operational Networking", ACM SIGCOMM 2003 Workshops, August 25&27, 2003, Karlsruhe, Germany. http://gravity.lbl.gov/grbell/

[12] J. Heinänen, F. Baker, W. Weiss and J. Wroclawski, "Assured Forwarding PHB Group", Request For Comments 2597, June 1999. http://www.ietf.org/rfc/rfc2597.txt?number=2597

[13] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–413, August 1993.

[14] K. Kilkki, *Differentiated Services for the Internet*, Macmillan Technical Publishing, Indianapolis, IN, USA, 1999, ISBN 1-57870-132-5.

[15] Nokia Research Center, "Nokia IPv6 Tutorial", September 2002.

[16] B. Davie, A. Charny, J.C.R. Bennett, K. Benson, J.Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu and D. Stiliadis, "An Expedited Forwarding PHB", Request For Comments 3246, March 2002. http://www.ietf.org/rfc/rfc3246.txt?number=3246

[17] N. Daswani, H. Garcia-Molina and Beverly Yang, "Open Problems in Data-Sharing Peer-to-Peer Systems", *Proceedings of the 9th International Conference on Database Theory (ICDT 2003)*, pp. 1–15, Siena, Italy, January 2003. Published by Springer-Verlag Heidelberg, Volume 2572 / 2003. Also available at: http://dbpubs.stanford.edu/pub/2003-1

[18] Gnutella: Distributed Information Sharing, 2000. http://gnutella.wego.com/

[19] K. Tutschku, "Management of Peer-to-Peer Networks", *2. Würzburger Workshop "IP Netzmanagement, IP Netzplannung und Optimierung"*, Würzburg, Germany, July 24, 2001. http://www3.informatik.uni-wuerzburg.de/ITG/2001/vortrag/vortrag13.pdf

[20] B. Yang, H. Garcia-Molina, "Improving search in Peer-to-Peer Systems", *Proceedings of the 28th International Conference on Distributed Computing Systems*, Vienna, Austria, July 2002.

[21] M. Schlosser, M. Sintek, S. Decker and W. Nejdl, "A Scalable and Ontology-Based P2P Infrastructure for Semantic Web Services", *Proceedings of the Second International Conference on Peer-to-Peer Computing (P2P'02)*, Linköping, Sweden, September 2002.

[22] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications", *Proceedings of ACM Sigcomm 2001*, pp. 149–160, San Diego, CA, USA, August 2001. http://www.acm.org/sigcomm/sigcomm2001/p12-stoica.pdf

[23] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham and S. Shenker, "Making Gnutella-like P2P Systems Scalable", *Proceedings of ACM Sigcomm 2003*, pp. 407–418, Karlsruhe, Germany, August 2003. http://www.acm.org/sigcomm/sigcomm2003/papers/p407-chawathe.pdf

[24] S. Saroiu, P.K. Gummadi, and S.D.A. Gribble, "Measurement Study of Peer-to-Peer File Sharing Systems", *Proceedings of Multimedia Computing and Networking 2002 (MMCN'02)*, San Jose, CA, USA, January 2002.

[25] Q. Lv, S. Ratnasamy and Scott Shenker, "Can Heterogeneity Make Gnutella Scalable?", *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS 2002)*, pp. 94–103, Cambridge, MA, USA, March 2002. Published by Springer-Verlag Heidelberg, Volume 2429 / 2002. Original version available at: http://www.cs.rice.edu/Conferences/IPTPS02/165.pdf

[26] Sharman Networks Ltd., "KaZaA Media Desktop", 2001. http://www.kazaa.com/

[27] Q. Lv, P. Cao, E. Cohen, K. Li and S. Shenker, "Search and Replication in Unstructured Peer-to-Peer Networks", *Proceedings of 16th ACM International Conference on Supercomputing (ICS'02)*, New York, NY, USA, June 2002.

[28] IRTF Peer-to-Peer Research Group. `http://www.irtf.org/charters/p2prg.html`

[29] B. Yeager, "[P2Prg] Proposed research tracks …", P2Prg mailing list archives, November 10, 2003. `https://www1.ietf.org/mail-archive/working-groups/p2prg/current/msg00120.html`

Economics of Peer-to-Peer
Music Industry Case Study

Marcin Matuszewski
Researcher
Networking Laboratory, HUT
Otakaari 5, 02150 Espoo, Finland
e-mail: marcin@netlab.hut.fi

Abstract

The world will be e-enabled and mobilized… Everything that can be digitalized will be digitalized (Anssi Vanjoki, Nokia, Executive Vice President). Those words perfectly present current situation in the music industry. Big stakeholders of the music industry try to oppose Peer-to-Peer (P2P) technology saying that it is music piracy that causes decrease in their revenue. The truth is that they try to slow down the development of Internet music retailing, looking at the same time for a viable business model that would secure their position in the new Internet world. Peer-to-peer networks represent the most efficient and a cost effective medium for trading of digitalized music. The fact is that the digitization of music necessitates a rethinking of their production and distribution economics.

This paper tries to analyze economic issues of online music distribution. It presents the influence Napster and its clones had on the music industry. It also conducts a study of the users of the file-sharing services. Based on our analysis we claim that p2p is not a threat, it is an opportunity.

1 Introduction

Napster launched the peer-to-peer revolution in October 1999. In just three days, over 4000 people downloaded the software and proved Napster's potential industry power. Its easy to use interface, that enabled access to unbounded free music resources, induced its widespread popularity and an extremely fast growth. The increasing number of its users provoked concern about the future of the music industry. It was a possible threat to labels and the Recording Industry Association of America (RIAA) that represents the music industry in the United States. A couple of months later RIAA sued Napster for the copyright infringement. The industry argued that this file-sharing service was contributing to massive copyright violations, as Napster users trade tens of thousands of songs every day. In February 2001 9th a Circuit Court decided that Napster violates the copyright law and ordered Napster to install filters and blocks to prevent transfer of copyrighted material. Napster trial has not finished music industry headache. New services such as Gnutella and Freenet, whose decentralized architecture makes more difficult to shut them down, have been launched. Napster has not only changed the conditions under which the copyright law is applied, but what is more important, it has altered the landscape of music retailing. New possible business models that have emerged together with free file-sharing services have caused big changes in the economics of music distribution. It was only the beginning of the Internet revolution in content distribution.

This paper is organized as follows. Section 2 presents the influence Napster and its clones had on the music industry. In section 3, we conduct the study of the users of music distribution services. Section 4 discusses properties of music distribution and presents a file-sharing service business model.

2 Peer-to-peer music industry threat

There are many possible factors that can influence the number of record sold. The major factors in the last decade could be record prices, income, economic situation and substitutes.

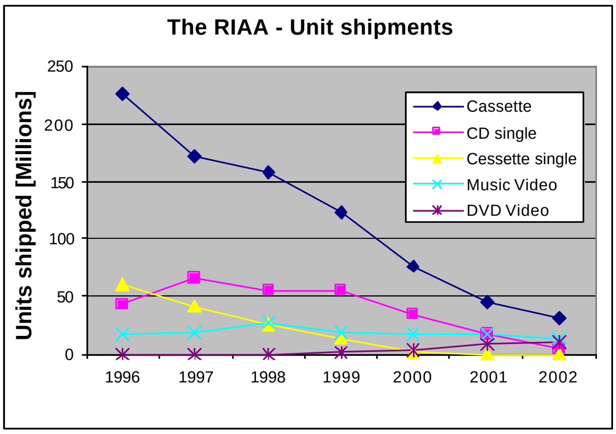


Figure 1. Unit shipments in 1996-2002 [36].

Year	**1997**	**1998**	**1999**	**2000**	**2001**	**2002**
U.S. Population [Thousands]	272912	276115	279295	282339	285024	288369
Gross Domestic Product (GDP) [Billions of Dollars]	8318.4	8781.5	9274.3	9824.6	10082.2	10446.2
Real Gross Domestic Product (GDP) [Chained (1996) BUSD]	8159.5	8508.9	8859	9191.4	9214.5	9439.9
Gross Domestic Product per capita change [%]	**5.20**	**4.34**	**4.41**	**4.79**	**1.65**	**2.41**
Personal Income [Billion of Dollars]	6937.0	7426.0	7777.3	8406.6	8685.3	8922.2
Personal Income per capita change [%]	**4.69**	**5.81**	**3.54**	**6.93**	**2.34**	**1.54**
Personal Expenditures Durable Goods [Billions of Dollars]	642.5	693.2	760.9	819.6	858.3	871.9
Real Personal Expenditures Durable Goods [Chained (1996) BUSD]	657.3	726.7	812.5	878.9	931.9	999.9
Personal Expenditures Durable Goods per capita change [%]	**2.98**	**6.64**	**8.52**	**6.55**	**3.74**	**0.41**
Inflation [%]	2.30	1.60	2.20	3.40	2.80	1.58
Unemployment Rate [%]	**4.90**	**4.50**	**4.20**	**4.00**	**4.80**	**5.80**
CDs Unit Shipment [Millions]	753.1	847	938.9	942.5	881.9	803.3
CDs Value [Millions of Dollars]	9915.1	11416	12816.3	13214.5	12909.4	12044.1
Price Per Unit	13.17	13.48	13.65	14.02	14.64	14.99
Price Change [%]	3.22	2.37	1.28	2.71	4.40	2.43
CDs Turnover Difference [Millions of Dollars]	-19.6	1500.9	1400.3	398.2	-305.1	-865.3
CDs Unit Shipment change [%]	-3.31	12.47	10.85	0.38	-6.43	-8.91
Price elasticity (arc elasticity)	**-1.06**	**5.00**	**8.11**	**0.14**	**-1.54**	**-3.89**

Table 1 Statistics of the U.S. economy [36,37,38,39,40,41].

According to the RIAA 2002 annual report [36], the number of units domestically shipped from record companies to retail outlets and special markets, like music clubs and mail order fell by 10.3% in 2001, and by subsequent 11.2% in 2002.

RIAA accuses mainly file-sharing services like Napster of those losses [2]. In this Section we will examine all of the pointed factors and we will try to answer the question, whether file-sharing services really had the main influence on this situation.

We will concentrate our analysis on CDs shipment values presented in the RIAA report [36]. This music industry product has experienced a decline in shipment one year after first peer-to-peer network had been launched. The shipment of the rest of the RIAA products according to Figure 1 has been declining for quite a long time. Therefore, even if it seems possible, that MP3 file sharing could have a negative effect on RIAA product shipments, it is hard to judge if it was the case. We will also exclude vinyl records from our analysis, because in our opinion any digital music products cannot have an influence on small vinyl records shipment. Records of this kind are purchased by music connoisseurs who are not sensitive to new music technologies.

2.1 Economic Recession

In this section we will try to perform an analysis that will help to show whether online music distribution is responsible for worse results of the music industry. We will also try to examine the most important reason of the current situation in that sector.

As far as music industry results are concerned, there has been a continuous growth of the main figures between 1997 and 2000 followed by a considerable fall in the next two years. It applies mostly to CDs unit shipment, which, after remaining almost unaltered in 2000, went down by 6.43 percent in 2001 and 8.91 percent in 2002, and to CDs turnover declining by 305.1 million dollars in 2001 and 865.3 million dollars in 2002.

In our opinion the economic downturn has been a major factor in the decrease of revenues in the music industry. According to the National Bureau of Economic Research (NBER), an official panel of senior economists, recession began in March 2001 [3].

The data in Table 1 clearly shows, that the growth rate of gross domestic product, both nominally and in real terms, was in 2001 much weaker than in the previous years. The difference is even more apparent, if we analyze the change of gross domestic product per capita, which, after the steady increase from 4.34 percent in 1998 to 4.79 percent in

2000, reached its lowest level at 1.65 percent in 2001 and increased only to 2.41 percent next year. As far as personal income and personal expenditures are concerned, the situation in 2001 and 2002 was even more dramatic. In 2001 the percentage change of personal income per capita slumped from 6.93 percent to 2.34 percent and continued to fall reaching its lowest at 1.54 percent in 2002. The percentage change of personal expenditures for durable goods per capita rapidly decreased from 6.55 percent in 2000 to 3.74 percent in 2001 and only 0.41 in 2002. Moreover the rate of unemployment reached the high level of 4.8 percent in 2001 and 5.8 percent in 2002.

Predictably the downturn in the economy is easier to notice by examining quarterly and monthly data. In the second quarter of 2001 gross domestic product was growing at a very weak 0.2 percent annual rate [11]. This was mainly the result of a recession in business profits and business spending, which contributed to companies' production and cost slashes. The US companies cut capital spending by 14.6 percent. These were the worst results since the second quarter of 1980. Companies' after tax profits decreased by 7.8 percent in the first quarter and by another 2 percent in the second quarter of 2001. In the third quarter of 2001 gross domestic product decreased by 0.4 percent [7].

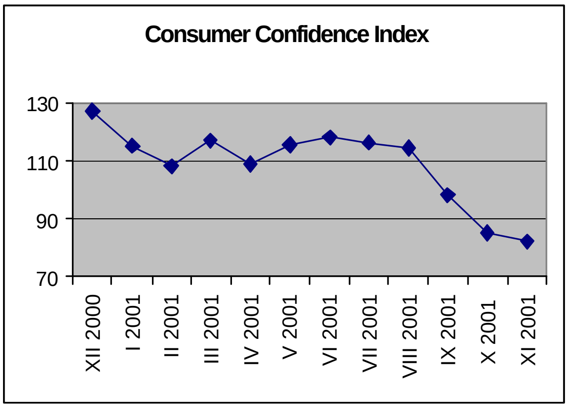


Figure 2. Consumer Confidence Index [9].

Due to considerable job cuts, which hit a ten-year high, the unemployment rate in August and September 2001 leveled out at 4.9 percent [9,10]. In addition, the September 11 attacks and war on terrorism, as well as the danger of further terrorist attacks, have taken their toll on consumers, who changed their attitude to life and started to look more to the future. The consumer spending increased by 2.5 percent in the second quarter and only by 1.2 percent in the third quarter of 2001 [6]. The economic downturn resulted in the fall of consumer confidence, which is measured by a consumer confidence index. The index shows the optimism of consumers regarding the economic situation. It is based on a survey of about 5000 households. 40 percent of the index reflects consumer opinion on current condition, the remaining 60 percent – on their future expectations.

2.2 Price elasticity

According to the "law of demand" the higher the price of the good the less customers will purchase. In order to estimate customer's demand economists evaluate the sensitivity of customers on price changes. The most widely adopted measure of the customer sensitivity to price is known as "price elasticity" on demand. It is defined as a ratio of the percentage change in quantity, divided by the percentage change in price. However, it is customary to compare absolute values of the ratio. The price elasticity of demand is equal to one, if a one percent drop in the price of a product causes a one percent increase in demand for the product.

Goods that are more essential to everyday life typically have lower elasticity. Goods with many substitutes or goods that are not essential have higher elasticity. Goods with the price elasticity below 1.0 are called inelastic goods meaning that customers are price insensitive. Food is the best example of an inelastic good. Goods with the price elasticity above 1.0 are called elastic goods. It means that customers are price-sensitive, so when prices rise, customers cut back on the quantity purchased. It is reasonable to expect that CDs are elastic goods, especially because there is a wide range of its legal substitutes like the radio, TV music programs, etc. Besides, according to the market studies presented in the following Sections, youths are spending almost the same time on listening to the radio as on listening to CDs.

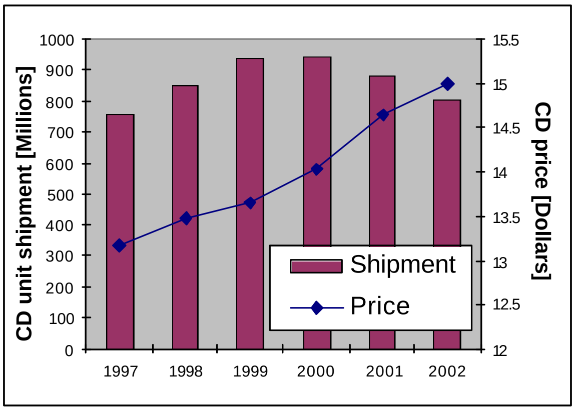


Figure 3. CD's price and shipment [36].

We have calculated CDs' price elasticity between 1997 and 2002. As an input of our calculation we have taken market data prepared by RIAA [35,36]. We found, that CDs have an average price elasticity equal to 2.82. Therefore, remembering that in years 2001 and 2002 U.S. has faced an economic recession and the personal expenditure on durable goods has decreased, this analysis can suggest that file-sharing services cannot be responsible for the whole decrease of revenues of the music industry. The poor economic situation as well as the pessimistic expectations of consumers have led to more selective spending which means that relatively expensive goods not believed to be desperately needed may remain outside the shopping basket.

Our conclusion has been supported by Randy Lennox of Universal Music Canada, who admitted that the Internet is not really the issue: "The real issue is the distribution of consumers' entertainment dollars. We have tons of evidence from surveys and market tests that have convinced us that people believe CDs are too expensive. If we reduce the retail price by 25–30 percent, we expect to see a corresponding increase in music consumption. That means more music in people's hands, and larger audiences for working musicians.” [34]. The Universal Music Group (UMG) that distributes almost a third of all recordings sold in the US announced at the beginning of September 2003, that it was introducing the new pricing policy lowering prices as much as $6 a disc. Aim of the new pricing was "bringing music fans back into retail stores and driving music sales."

3 Market analysis

Successful companies operating in the Internet must understand customers’ needs, their behavior and the reasons behind purchasing. They must identify users’ pain and find a solution to help them. Available surveys and market research reports are indispensable sources of such information. In this section we will try to touch those issues.

3.1 Market growth

Nearly one-third of the U.S. population of 12 years old and over has ever downloaded music from the Internet. For the purpose of our further analysis those people will be called downloaders. According to Figure 4 the number of downloaders represents a stable, almost twenty-five percent annual growth.

Youngsters, between 12 and 17 years old, represent the fastest growing group [17]. One out of two (52%) U.S. youngsters have ever downloaded music from the Internet whereas 32 % have done it in the last month (see Table 2). Older people are more skeptical about Internet music distribution. They tried it, but they do not feel enthusiasm for it. Gender break [17] from December 2002 presents that men (26%) are more likely to download music from an online file-sharing service than women (12%). More men (13%) than women (6%) have downloaded music in the last 30 days.

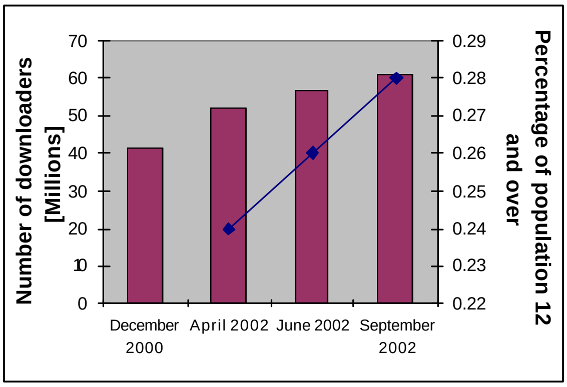


Figure 4. Downloaders group growth [18,19,20,21].

Age	April 2002		December 2002	
	Ever	In last 30 days	Ever	In last 30 days
12-17	41%	23%	52%	32%
18-24	45%	26%	44%	24%
25-34	26%	11%	23%	8%
35-54	14%	4%	12%	5%
55+	2%	1%	3%	1%

Table 2. Age break of downloaders [17,18,19,20,21].

According to another study [16] from 2001, young men downloaded music four times per month, whereas young women only three times. The gender difference is not surprising. Typically young men are much more eager to use cutting-edge technologies than women. The difference can also be noticed in the mean number of times both genders spent on listening to music in the last 30 days [16].

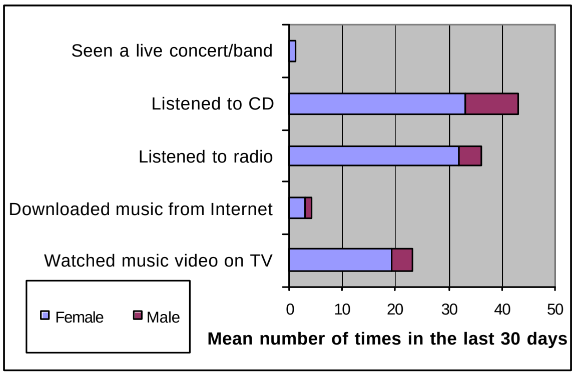


Figure 5. Music listening habits of youth, 2001 [16].

Figure 5 presents results of the survey conducted in the first quarter of 2001. The results show that traditional sources of music, such as radio and TV, are still much more popular than downloading music.

3.2 Reasons for using p2p networks

According to the survey conducted by IDC [13], Napster users indicated that choice and convenience, rather than price, are the key drivers of peer-to-peer systems. The result of the research, that is presented in Figure 6, shows that over one-tenth of respondents are downloading music from P2P file sharing networks because it is more convenient than purchasing it from the traditional retailers. At the same time twenty four percent believe this is only one possibility of getting a specific song that they otherwise would not buy. This can be interpreted that users want to choose one specific song from the album instead of buying the whole one. Therefore record companies and online retailers should offer multiple price points for single tracks and albums.

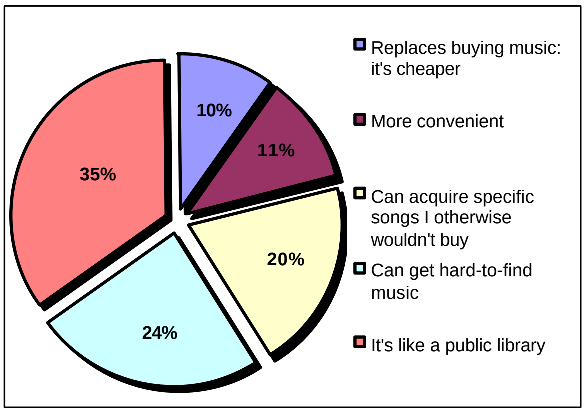


Figure 6. Reasons for using Napster [13].

Nearly two-fifth (24%) of users value the possibility of downloading music, that is very hard to get, like limited editions, as well as music produced by small labels. Overall, majority of people mostly value the convenience and simplicity of online file sharing systems that allow them to save scarce time. The rest of the results suggest that current prices of music content are too high and therefore customers are switching to other, cheaper music sources like peer-to-peer file sharing networks. This is another evidence in support of the analysis conducted in the second section.

3.3 Effect of downloading on CD purchase

According to another market study conducted by Ipsos [19] in December 2002 nearly three-quarters (73%) of music downloaders in U.S. reported that they have downloaded music in order to sample it before decision about purchasing. The results of another survey [17,18,19,20,21] conducted earlier in 2002 are presented in Table 3. The results confirm those findings.

Has your CD purchasing changed?	Heavy Downloaders	Light Downloaders	All Downloaders
Increased	26%	27%	27%
Stayed the same	48%	61%	58%
Decreased	26%	12%	15%

Table 3. Effect of Downloading on CD Purchases [17,18,19,20,21].

This data presents the effect of downloading on CD purchase by music downloaders aged 12 and over. Downloaders have been divided into two groups: first heavy downloaders who download music from the Internet several times a week and second light downloaders who do it less frequently. According to what we can see on the left-hand side of the table, the number of heavy downloaders who have been encouraged to buy more CDs after they downloaded MP3 files from the Internet and the number of downloaders who have decreased their willingness to buy legal music are equal. The second group becomes more likely to buy CDs after using p2p file sharing. Those findings can be a suggestion for retailers that people want to listen to the sample of good quality music before they make a decision about purchasing CDs. Similar results have been obtained by Jupiter Research. They found that people at the ages of 18 to 24 who spent less than $20 on music within a three-month period were likely to remain at a constant purchasing level despite online music use. All the others agreed that they had increased spending as a result of online music use.

3.4 Market simulation

A market simulation research [17,18,19,20,21] conducted in July 2002 revels that 27 percent of downloaders aged 12 and over reported having paid for a music downloaded from the Internet. In September 2002 this number grew to 31%.

The results of the July 2002 research presented in Figure 7 show that almost 2 out of 10 music downloaders prefer pay-per-download option in contrast to only eight percent who opt for a subscription based service. This is a surprising finding especially because according to previous research [27,28,29], users of the Internet and telecommunication services prefer flat rate charging. Besides, if no pay-per-download option were possible, the percent of downloaders that have reported having paid for music downloaded would decrease to 12%.

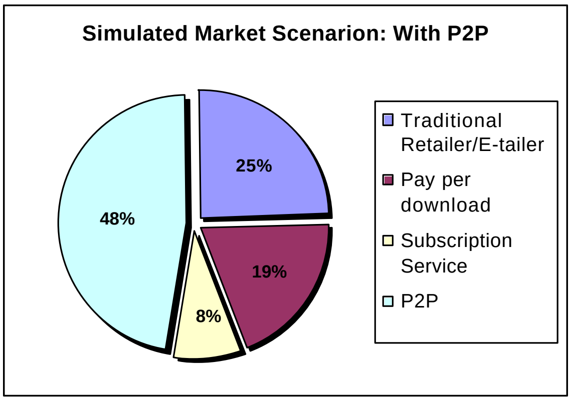


Figure 7. Downl oaders preferences for obtaining music [20,21].

Research also suggests that not all of current downloaders would move back to the traditional retailing channel if peer-to-peer options were removed. The majority of downloaders would continue buying music in digital format from the Internet. As before the pay-per-download option would dominate among digital music purchasing methods. Only 23% of downloaders would change their purchasing habits choosing traditional retailers. It is worth to point out that youngsters between 12 and 17 years old are more likely to pay for music than their older colleagues. The possible explanation of this finding is that youngsters have grown up in the Napster era and therefore music in the form of MP3 files is natural for them. They recognize the value of this music.

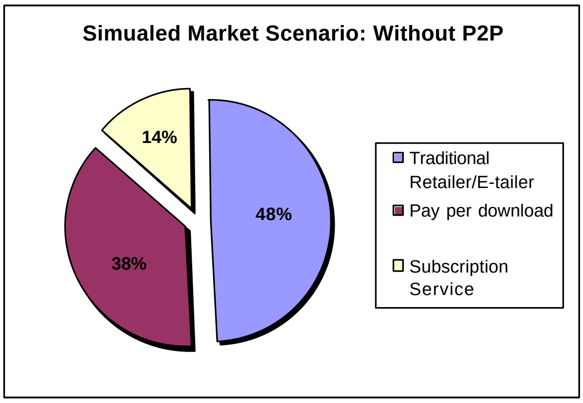


Figure 8. Potential downloaders' preference. P2P not available [20,21].

3.5 Sociology of music downloading

The sociology of music downloader activity is not less interesting. Most of the downloaders believe that there is nothing wrong with their actions [19], whereas nearly forty percent (39%) stated that copying music in order to give it to friends is all right. In contrast, one out of ten (9%) agreed that their behavior is wrong and only one-in-five (21%) believe, that free music downloading hurts artists. This data clearly shows that the majority of Internet music fans are not bothered about sociological aspect of their actions. They interpret intellectual property rights in their own terms.

4 Business Model

The most essential advantages of the peer-to-peer retailing business model are scalability, much easier and more effective marketing and less expensive retailing and distribution. These benefits help to cut costs and can lead to a wider selection of music available at lower prices.

4.1 Distribution and Retailing

The advances in the information and communication technology enable economic agents to interact directly. Virtual transactions over the Internet overcome the physical distance between people. The recent economy trends suggest that, by using the Internet connectivity, companies can remove middlemen from the value chain. There are strong grounds for this trend. According to IDC research [15], the distribution and retailing process absorbs 30 percent of revenue from Audio CD sales in contrast to the 8 percent revenue received by artists and publishers.

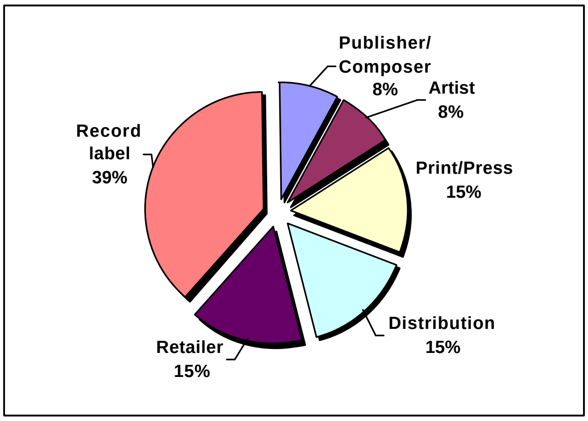


Figure 9. Distribution of revenues from audio CD sales [15].

With the help of the Internet, distribution and production costs can be cut. Therefore, music can be offered at much lower prices and that according to our previous study can potentially increase the audience. It also opens the possibility for small labels to compete with the Big5, something that is almost impossible in traditional retailing. Therefore, a much simpler value chain and much lower costs can lead to a much wider music selection. The file-sharing services enable a song to be downloaded and charged individually in a very cost efficient manner. As it was presented in the previous section, 20 percent of users of file-sharing services prefer this method of purchasing.

4.2 Bundling

Bundling is the practice of joining related products together for the purpose of selling them as a single unit. Bundling arrangements usually feature special pricing arrangements that make it cheaper to buy the products and services as a bundle, than separately. Because of the high distribution and production cost, the music works are mostly bundled together. Each album contains several tracks from the same or different artists.

A file-sharing system also has potential to introduce new ways of bundling. It is possible to bundle a work from a certain group of artists based on archived previous transactions. Another possibility is to offer to the user an unlimited access to a certain pool of music works.

4.3 Online marketing

Music is an experience good and therefore it must be experienced before anyone wants to purchase it. This is a well-known rule that exists als o in the Internet domain, as was clearly presented in the previous sections. Huge amounts of money are spent on marketing and promotion for a particular album. Music is promoted on the radio, television and during concerts. It is worth to mention that the last option is quite controversial nowadays. Up to now, labels have not received any revenue from concerts. Life performance rules have not changed since the earliest music market where music users could directly pay music producers. Seeing their revenue dramatically falling, the labels have recently decided to change those rules. They claimed that they have rights to a part of the concerts revenue because they invest in an artist's promotion.

Giving customers a possibility to download music before they buy it can be a very powerful marketing tool as has been proved by Ted Cohen, a senior vice president of New Media for EMI who opposed [31] the Red Hot Chilli Peppers and Metallica claims that the possibility of buying particular tracks is "destroying the album". He gave the example from iTunes, an online music retailing system launched by Apple in April 2003: "In the first days of iTunes availability 10 of the top 50 tracks were Coldplay. However, the album was not in the top 50. In the third week the album moved into the top 10 and half of the tracks moved out of the top 50". The conclusion we can get from this example is that when people discover that there are more than two good songs on the album, they started to believe that it is worth to buy the whole album.

Legal online music retailers and distributors must start their operations by building customer trust. An example of a wrong strategy can be presented analyzing the case of iTunes [31] that has removed network functionality from their service. This functionality allowed customers who bought music, to share the files across the network. That action did not bring the intended goal. It only decreased customer confidence, especially in a situation when they still had free music distribution networks as an alternative. As a result, a hack that restored the functionality has been quickly compiled. Letting people share in a controlled way their legally purchased music has strong marketing power. Customers, who share music with friends, become marketers and open new channels.

4.4 Target marketing

P2P retailing systems enable target marketing that was quite difficult and expensive with traditional music retailing methods. The market segmentation is much easier. Labels can distinguish the major market segments, target one or more of these segments and develop products and marketing programs tailored to each selected segment. Customer Relationship Management (CRM) is the key element of the marketing machine. Based on information stored in the CRM system, like the previous transactions, the music label can easily segment the market. A CRM used for Internet transactions makes discovering customer's needs much easier and helps to create offerings that respond to these needs. The label can adjust product prices and efficiently advertise to reach the target market. Based on banners in the p2p software user interface it is possible to offer users tailored music products. Bundling can be also effectively connected with direct marketing. As music must be experienced before customer decides to purchase, new songs of the user's favorite artist can be easily bundled together with a piece of music recorded by a new artist from the same genre.

4.5 Superdistribution

In order to make the p2p retailing business model viable, the file-sharing technology should be accompanied by legal and business process advances. One of the most advanced concepts of intellectual property protection for the digital age is Digital Rights Management (DRM) called Superdistribution. DRM allows content owners to securely attach the right management information to each piece of distributed content. Retailers can easily determine the conditions of the transaction, like price, subscription package or just pay-per-run option. DRM benefits also the customer who can pay based on his or her own preferences. Let us take, as an example, file sharing among a group of friends. One group member buys a song of a recently discovered group. He likes the song and decides to share it with friends. Based on DRM information attached to the file or downloaded automatically from the Internet, his friends are allowed to play this song n-times for free and finally decide that this song or even album is worth to purchase.

The development of DRM systems is, however, still on-going. There are many attempts around the world to make this system working. Recently, a company called Beep Science demonstrated a life mobile DRM solution on ITU Telecom World 2003 in Geneva. Another initiative to facilitate the distribution of legitimate content is The Content Reference Forum (CRF). CRF promotes the adoption of specifications and design guidelines, leveraging existing standards to create an open framework for interoperable, platform- and business model-independent digital content distribution. The key members of this forum are companies like Microsoft Corporation, Universal Music Group and VeriSign, Inc. The forum has already published a candidate specification [12] and plans to make a trial at the beginning of 2004.

4.6 Scalability

According to David P. Reed [32], peer-to-peer networks can be classified as Group forming Networks (GFNs). GFNs have an important network capability that directly supports affiliations among subsets of its customers. The number of groups that can be formed in these networks can potentially grow exponentially with N according to formula 2^N , where N is the number of group members. According to David Reed, the exponential law of GFNs creates increasing returns as scale increases. Kalervo and Kilkki have modified Reed's formula and have shown that the GFN's value increases exponentially when the number of customers approaches 90 percent of the population defined as all of the potential customers that can use the service. When two group forming networks merge, substantial new value is created $2^{M+N} = 2^M 2^N$.

The EBay auction system is a good example of a GFN. Their concept is to encourage its members to set up specialized auctions on the eBay web site. Each auction can be treated as a single group of a seller and potential buyers. Their business model is based on small fees paid by each customer that want to sell his or her product. Economic results of this company prove the scalability of GFNs. In 1998, eBay had 1 million shoppers that set up 600,000 items for sale, and generated $6 million in revenue. By 2000 there were 3 million items for sale. In 2001, the company hosted $5 billion worth of transactions.

4.7 Napster case

In February 2001, Napster became the 13th most visited web site with 16.9 million unique visitors. Despite its success Napster had never generated money from its operation. The report conducted by Jupiter Reports showed that 68 percent of 40 million users in January 2001 would be willing to pay $15 per month for the Napster service. If Napster lowered the suggested monthly fee to $4.95, the number of users that would stay with Napster would grow up to 81 percent. This data presents Napster's strength – very strong brand.

The new Napster's management that was elected by its new stakeholder Bertelsmann, the world's third-largest media company, owner of the BMG record label, decided to reincarnate Napster as a revenue-generating legitimate business. The management felt that it can succeed with the subscription based model. They suggested that a basic service with a limited number of transfers would cost somewhere between $2.95 and $4.95 a month, whereas an unlimited access service would cost between $5.95 and $9.95. Operational expenses included rights to play-lists from record companies, overhead for billing and customer service, technology development for a security standard to prevent songs from being passed around, record companies fees per song, and songwriter fees per song [1].

In February 2001, Napster CEO and Bertelsmann's head of e-commerce Hank Barry offered $150 million per year to the five major record companies and $50 million to the independent ones [4]. He explained that if the company kept only 2 million of its 64 million registered users, it would make about $119 million per year based on an average payment of $4.95 a month. If that subscriber base grew to 14 million paying customers, the revenue would reach $832 million a year. A fee of one billion dollars paid over the next 5 years to labels would correspond to $5.4 billion in CDs since the labels would have no additional production and distribution costs associated with this fee. This proposal was however rejected by the labels. Finally, the second version of Napster was launched in October 29th 2003. It offers users a digital music library of 500,000 songs. Clients can choose both an a la carte store and a premium subscription service. The former option offers $0.95 per track payment, whereas the latter an unlimited streaming and downloading for $9.95 per month [26].

5 Conclusion

In 2002, the number of Internet music downloaders has reached almost 30 percent of U.S. population. Music downloading activity became more and more popular especially among young people that have grown up in the Napster era. They appreciate the possibility of getting music from the Internet. They understand its value. The presented market analysis suggests that the almost 25 percent annual growth of downloader population that was observed in previous years, can even accelerate when young people get older.

Our analysis also suggests that file-sharing services cannot be responsible for the whole decrease of revenues of the music industry. In years 2001 and 2002 the United States has faced economic recession that has resulted, among others things, in the decrease of personal expenditure on durable goods. The poor economic situation as well as the negative expectations of consumers have caused them to shop more selectively. It means that durable goods, like CDs, that are not believed to be desperately needed may remain outside the shopping basket.

According to data presented in this paper it seems that peer-to-peer systems have altered the landscape of music retailing and distribution. Current peer-to-peer music downloaders expect simple pay-per-download service that offers a wide range of music choice. They also want to have a possibility to pay for particular songs in addition to the traditional whole album purchasing method.

The new possible business models that have emerged together with free file-sharing services are causing big changes in the economics of music distribution. This paper has studied possible business models and tried to answer the question if online music distribution is just piracy or maybe the new way of very efficient music retailing. Our analysis suggests that scalability, much easier and more effective marketing, less expensive retailing and distribution are all advantages of file-sharing services.

6 References

[1] Afuah Allan, Tucci Christopher L.: Internet Business Models and Strategies, McGraw-Hill, 2003

[2] BBC News: Piracy blamed for CD sales slump, http://news.bbc.co.uk/1/hi/entertainment/new_media/1841768.stm, accessed November 5, 2003

[3] Business Cycle Dating Committee, National Bureau of Economic Research: The NBER's Business-Cycle Dating Procedure, http://www.nber.org/cycles/november2001/, accessed November 8, 2003

[4] CNET NEWS.com: Napster offers recording industry $1 billion, http://news.com.com/2100-1023-252862.html?legacy=cnet, accessed November 21, 2003

[5] CNET NEWS.com: Start-up to fuse file swapping with e-commerce, http://news.com.com/2102-1017_3-243438.html, accessed November 21, 2003

[6] CNN Money, U.S. economy shrinks, http://money.cnn.com/2001/10/31/economy/economy/, accessed November 9, 2003

[7] CNN Money: Economists call it recession, http://money.cnn.com/2001/11/26/economy/recession/, accessed November 9, 2003

[8] CNN Money: Napster's Back, October 29, 2003 http://money.cnn.com/services/tickerheadlines/prn/law058.P2.10292003010215.22783.htm, accessed November 21, 2003

[9] CNN Money: Short recession likely http://money.cnn.com/2001/09/20/economy/toll_economy/, accessed November 9, 2003

[10] CNN Money: U.S. job cuts soar, http://money.cnn.com/2001/10/05/economy/econo

my/, accessed November 9, 2003

[11] CNN.com: Peter Viles: Economic numbers tell a murky tale, http://edition.cnn.com/2001/US/08/29/viles.debrief.otsc/, accessed November 9, 2003

[12] Content Reference Forum: Candidate Specification Public Review, http://www.crforum.org/articles/about/candidate.html, accessed December 5, 2003

[13] IDC: Reasons for Using Napster, IDC #24454, April 2001, http://www.idc.com, accessed November 20, 2003

[14] IDC: U.S. Digital Music Download Sales 1999-2004, IDC #22632, July 2000, http://www.idc.com, accessed November 20, 2003

[15] IDC: U.S. Distribution of Revenues from Audio CD Sales, IDC #22632, July 2000, http://www.idc.com, accessed November 20, 2003

[16] Ipsos-Reid: Online Music Distribution: Revolutionary Aspirations, First Quarter, 2001, http://www.ipsos-reid.com/pdf/publicat/wm.01.01.06.pdf, accessed October 25, 2003

[17] Ipsos-Reid: TEMPO: Keeping Pace With Digital Music Behavior, February, 2003, http://www.ipsos-pa.com/dsp_displaypr.prnt.cfm?ID_to_view=1743, accessed October 25, 2003

[18] Ipsos-Reid: TEMPO: Keeping Pace With Digital Music Behavior, December, 2002, http://www.ipsos-pa.com/dsp_displaypr.prnt.cfm?ID_to_view=1685, accessed October 25, 2003

[19] Ipsos-Reid: TEMPO: Keeping Pace With Digital Music Behavior, March, 2003, http://www.ipsos-pa.com/dsp_displaypr.prnt.cfm?ID_to_view=1763, accessed October 25, 2003

[20] Ipsos-Reid: TEMPO: Keeping Pace With Digital Music Behavior, September, 2002, http://www.ipsos-pa.com/dsp_displaypr.prnt.cfm?ID_to_view=1631, accessed October 25, 2003

[21] Ipsos-Reid: TEMPO: Keeping Pace With Digital Music Distribution. Tempo program overview. February 2003, http://onstage1.webex.com/onstage1/tool/docshow/enter.php?AT=Show&Type=Playback&ClientName=webex&FileName=http://onstage1.webex.com/seminar/414/play/276933586/matt.wrf&Rnd=657062298, accessed November 1, 2003

[22] Kotler, P.: Marketing Management, Analysis, Planning, Implementation, and Control, Eleventh Edition, Prentice Hall International Edition, 2003

[23] Liebowitz Stan: Will MP3 downloads Annihilate the Record Industry? The Evidence so Far, School of Management, Univ. of Texas at Dallas, 2003

[24] Lipsey, Richard G.: Macroeconomics, New York HarperCollins 1996

[25] Mullins John W.: The New Business Road Test, Prentice Hall, 2003

[26] Napster: Napster fact sheet, http://www.napster.com/facts.html, accessed November 21, 2003

[27] Odlyzko Andrew: Content is not king, AT&T Labs –Research, 2001

[28] Odlyzko Andrew: Internet pricing and the history of communications, AT&T Labs–Research, 2001

[29] Odlyzko Andrew: Internet traffic growth: Sources and implications, University of Minnesota, 2003

[30] Pindyck, Rober S.: Microeconomics, New York (NY) Macmillan 1992

[31] Raisford Miriam: Network Effect: Stan Liebowitz and the MP3 Debate, O'Reilly Network, www.openp2p.com/lpt/a/4038, accessed October 25, 2003

[32] Reed David P.: That Sneaky Expotential – Beyond Metcalfe's Law to the Power of Community Building, http://www.reed.com/Papers/GFN/reedslaw.html

[33] Senator Binden Joseph R.: Theft of American Intellectual Property: Fighting Crime Abroad and at Home, U.S. Senate, February 12, 2002

[34] Stereophile: CD Price Drop, www.stereophile.com/news/11730/index.html, visited November 24, 2003

[35] The RIAA: 2002 Consumer Profile, www.riaa.com, accessed November 5, 2003

[36] The RIAA: 2002 Yearend Statistics, www.riaa.com, accessed November 5, 2003

[37] The White House: Economic Report of the President, U.S., February 2003, http://www.whitehouse.gov/ , accessed November 22, 2003

[38] U.S. Bureau of Labor Statistics: Monthly Labor Review and Handbook of Labor Statistics, periodic, http://www.bls.gov/ , accessed November 22, 2003

[39] U.S. Census Bureau: Motion Picture and Sound Recording Industries (NAICS 512): Estimated Revenue and Inventories http://www.census.gov/svsd/www/sas512.pdf, accessed November 23, 2003

[40] U.S. Census Bureau: Statistical Abstract of the U.S. 2002, http://www.census.gov/statab/www/ , accessed November 23, 2003

[41] U.S. Department of Commerce, Bureau of Economic Analysis: http://www.bea.doc.gov/bea/dn/home/gdp.htm, accessed November 3, 2003

[42] Zhang Michael X.: A Review of Economic Properties of Music Distribution, Sloan School of Management, MIT, 2002

Legal Issues in P2P Systems

Klaus Nieminen
Networking Laboratory
Helsinki University of Technology
Klaus.Nieminen@hut.fi

Abstract

This paper discusses legal issues related to peer-to-peer (P2P) systems. It describes the principles of copyright legislation including the international copyright treaties and the U.S. and European copyright legislations. Also the related privacy issues are covered. The Internet and P2P file sharing applications have enabled totally new possibilities for users, such as a free and independent communication channel and a fast and easy mechanism for file sharing. These new enablers have changed the rules in many ways, which have forced the legislatures to adapt the legislation to respond to the new environment.
Also the changing legislation is directing the P2P system development because aggressive litigation, new laws and interpretations have caused a lot of troubles for the P2P community. Thus, the trend seems to be towards smaller, closed P2P groups and global systems with more limited access to the content.

Key words: peer-to-peer, copyright, privacy, legislation

1 Introduction

During the past few years Internet users have begun to increasingly utilise the peer-to-peer communication model. Unlike the client/server model used by many popular Internet applications, peer-to-peer applications can act both as a client and a server. This capability enables P2P systems to work without any centralised infrastructure in an ad-hoc manner.

P2P applications do not require any centralised management or control. Therefore, it is extremely hard to control their usage, as would be required, e.g., in the case of lawful interception or preventing copyright infringements. Especially the interpretation of copyright laws raises many legal concerns but also the privacy and security issues need to be taken into account.

Even though the legal problems caused by the use of P2P applications have gained a lot of publicity lately, the reader must note that P2P is not a new invention. In fact, the Internet was originally a P2P network and the client/server model emerged truly only with the rise of the commercial Internet in the early 1990s.

1.1 Peer-to-peer Systems

In a P2P system the nodes have typically equal capabilities and any of them is able to initiate a session. P2P systems do not require any centralised infrastructure, although the peer-to-peer architecture does not prevent the use of this kind of infrastructure either. The different communication architectures are further elaborated in Figure 1.

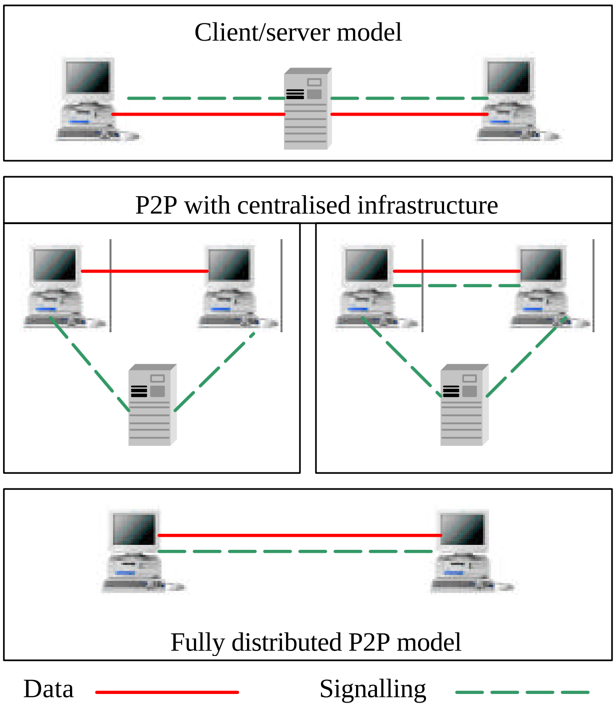

Client/server model
P2P with centralised infrastructure
Fully distributed P2P model
Data
Signalling

Figure 1: Communication models

Many P2P communication applications use centralised servers, e.g., for managing and distributing location, presence or registration information. SIP-telephony and instant messaging are good examples of systems that operate in this fashion.

For the large audience, the term P2P has recently become a synonym for file sharing applications that use the Internet to exchange files either directly between peers or using a media server as an intermediary.

Internet file sharing applications can connect to each other directly for downloading and uploading files. However, the problem is to find the peer from whom to retrieve the file. Many peer-to-peer applications solve this problem by introducing some centralised infrastructure. Two implementation examples are presented below.

- Napster uses a central server to store the index of all the files available within the Napster user community [1].
- Kazaa [2] uses a concept called supernodes for storing the file lists. Kazaa maintains a list of supernodes and information about logged users. For more information, see Section 1.2.

In addition, also the user group forums and news groups are occasionally referred to as peer-to-peer systems, even though they are typically client/server systems with a centralised server hosting the service [3]. From service users' point of view, only the content is contributed on a peer-to-peer basis but at the system level also the servers can form a peer-to-peer network, such is the case, e.g., with the USENET servers.

The domain name system is another good example of a system combining peer-to-peer networking with a hierarchical model of information ownership. The services like these do not include any special P2P related legal issues and they are lined out of this study. This paper concentrates on file sharing systems and the following section presents a case study of the most popular file sharing application, Kazaa.

1.2 Case Study: Kazaa

Kazaa is currently the most popular file sharing application with millions of users. At 9 pm on the 15th of November 2003 the figures were the following: 4 360 951 users online, sharing 766 949 385 files (5 643 136 gigabytes).

Similar to many other peer-to-peer applications, also Kazaa is used to distribute huge amounts of copyrighted material. Therefore, it is important to understand the mechanism how these applications operate.

In Kazaa one can search files by querying supernodes that are appointed from the nodes participating in the Kazaa network. Neighbouring nodes use supernodes to make queries and to upload the list of files they are sharing. Supernodes connect to other supernodes and thus form a distributed file list of shared files. [4]

When launched, a Kazaa application logs-in to a central log-in server operated by Kazaa. However, the application does not essentially need this connection because even if the log-in server is not available, the application starts making queries based on a list of supernodes hard coded into the application software. The list is updated when connecting to any supernode. [4]

If the application cannot reach any of the supernodes it knows, it connects to another server controlled by Kazaa to obtain a list of current supernodes [4]. This means that even though Kazaa does not know what files are shared, it has to maintain a dynamic list of current supernodes.

In P2P networking files can be searched based on many different parameters. For example in Kazaa [5], queries can be made using pre-defined file types. Each file type has its own set of parameters including different categories. Table 1 presents the different file types and categories used in Kazaa.

Table 1: File types and categories in Kazaa

File Type	Categories	Examples
Audio	115	Celtic, Rock, Ska
Documents	30	Cooking, Diaries
Images	17	Art, Erotic, Family
Playlists	115	same as audio
Software	11	Drivers, Games, OS
Video	21	Drama, Series, War

For example, when searching audio files other parameters include title, artist, album, language, year, quality, integrity and size. As can be seen from the example, the Kazaa applications are able to share and search any kind of digital content while the well-defined parameters make it easy for the user to find exactly the piece of content he is after.

A feature called participation level that is calculated for each user as a ratio of uploaded and downloaded megabytes, helps to boost the amount and quality of shared files. Users are encouraged to share more files because the users with a higher participation level get a better downloading priority and are able to do more queries.

2 Legal Issues Concerning the P2P Systems

No especially peer-to-peer centric legislation exists. Nevertheless, all other relevant laws, such as communication, competition, security, patent and privacy legislation have an effect on the development of P2P communication and developer business.

However, in this paper these topics are discussed only when they have clear interactions with copyright that has been identified as the main legal issue concerning P2P file sharing, and therefore also all peer-to-peer systems. Also lawful interception is briefly studied as an interesting problem in P2P communication.

2.1 Lawful Interception

Lawful interception (LI) is legally authorised official access to private communications such as telephone calls or email-messages [6]. Typically, the information is provided to a law enforcement monitoring facility by network operators, access providers or service operators. These parties need to make sure that the targeted party cannot detect the interception and that unauthorized personnel must not gain knowledge of interceptions or be able to perform LI.

In traditional fixed and mobile telephony networks, the LI functionality is well specified and therefore relatively easy to implement. The problem in peer-to-peer communication is that there is no certain point of interception that the traffic has to cross. In addition, it is much harder to identify the target traffic in P2P environment due to the larger variety of different communication services and identifiers.

Internet service users may also apply additional protection mechanisms such as virtual private networks or encryption that make interception even more difficult. The problem is mostly technical and the standardisation bodies including IETF, 3GPP and ETSI are working to solve the problem.

The LI legislation differs from country to country. Therefore only the principles and guidelines of P2P related LI legislation are discussed in this paper. It is especially essential to define the rules for lawful interception because at least in constitutional states police operations should be clearly regulated. The main principles include, e.g., the definition of:

- The cases in which the police can request LI,
- The procedures to obtain the permission for LI,
- How LI target is defined and what kind of information can be obtained by LI.

Concerning peer-to-peer communication, especially the definition of LI target is very important. With the PSTN the police needed only a permission to intercept calls to and from a certain number. In the IP world the police may need a right to intercept communication to and from a certain fixed or dynamic IP address, device or communication address and even these may not essentially cover every possible case, like P2P voice calls or instant messages from a library computer using a random source name.

The problem is that the legislation needs to give the police enough rights to do their work but in a way that does not risk the privacy of fellow citizens. Also the cost and harm caused by the interception should be proportional to achieved benefits. Therefore, even though technology makes something possible, it may not be feasible to implement it in practice.

2.2 Copyright

Copyright is an exclusive right to make and distribute copies, prepare derivative works and perform and display the work in public. The protected work has to be genuine and distinct and the international copyright becomes automatically to be the property of the author when the work is created.

Copyright is granted to the author under each national law. The national laws of individual countries are linked by the following international treaties administered by the World Intellectual Property Organization (WIPO): [7]

- Berne Convention,
- Brussels Convention,
- Geneva Convention,
- Rome Convention,
- WIPO Copyright Treaty,
- WIPO Performances and Phonograms Treaty.

These international treaties ensure that at least a minimum level of rights will be granted to authors in all contracting countries. Concerning P2P file sharing the most important treaties include the Berne Convention and the WIPO Copyright Treaty (WCT).

The Berne Convention [8] introduced the international copyright without registration in 1886. It has been signed by 151 states and it guarantees the minimum rights for the minimum duration of author's lifetime plus 50 years.

The WIPO Copyright Treaty [9] extends the copyright protection offered by the Berne Convention and GATT Trade Related Aspects of Intellectual Property Rights (TRIPS) Agreement to address the digital content issues. The WCT is applied to computer programs, compilations of data, cinematographic works and sound recordings. The WCT has currently been signed only by 42 countries.

P2P Related Copyright Issues
Considering copyright issues, already digitalisation combined with the global distribution medium offered by the Internet has caused many problems. The Internet has expanded much faster than the laws have been able to respond. Now the legislatures and courts are trying to catch up to protect intellectual property rights (IPR) while still maintaining the free flow of information over the Internet.
Peer-to-peer applications have made the situation even worse by introducing dedicated search engines and stronger anonymity. The copyright infringements are much harder to spot due to the anonymous and temporary file sources. Also the participation feature that is built into the P2P file sharing applications to prevent free riding has increased the amount of shared copyrighted files.

The recording industry and especially its spokesman in the U.S., the Recording Industry Association of America (RIAA), has tried to prevent the P2P file sharing by using or sponsoring the following juridical and technical methods:

- Technical methods include spoofing P2P networks with low quality or damaged music files or placing harmful or tracing code to the shared files. The code can, e.g., lock-up or slow down the computer or obtain the IP address and/or the username of the downloading user as well as the name of the downloaded file and the used program [10].
- Juridical methods include sending subpoenas to ISPs to obtain names of P2P users or to intimidate them to stop the activity. The RIAA is also suing the P2P application developers, service providers and the users caught from copyright infringements.

These actions, such as placing malicious code into the shared files and sending over broad subpoenas can also be illegal breaches of security and privacy. Therefore, there are organisations, such as the Electronic Frontier Foundation (EFF) that, e.g., initiates and defends court cases preserving individuals' rights. The RIAA, EFF and P2P United will be discussed in more detail in Section 3.2 American Groups Acting For or Against P2P File Sharing.

Despite many encountered problems, the RIAA's campaign seems to work and the use of peer-to-peer networks has clearly decreased during the last few months [11]. However, the chase can also just drive the users from open to closed peer-to-peer systems that are much harder or even impossible to control.

2.3 Entities at Risk of Legal Liability

Not only the users of P2P file sharing applications who are caught infringing a copyright law can be considered liable for copyright infringement. Also the employers and Internet service providers (ISPs) of these users can be considered to be liable in certain circumstances. In addition also the P2P file sharing technology and application developers may be subpoenaed for many reasons. The risks of these individual groups are further studied below.

P2P users
If caught violating copyright laws, the P2P application user may be sued to court and for example, in the U.S. the implications of copyright violation may be quite dramatic. The maximum penalties from such violation are $150 000 per instance, punitive cash damages and a possibility of being imprisoned.

Depending on the national law, only the unauthorised uploading of copyrighted content may be illegal or as in the U.S., also the downloading of such content is prohibited.

Employers
The RIAA has also begun to seek legal actions against companies that do not prevent their employees to infringe copyright laws by sharing files in P2P networks using work computers and Internet connections.

For example, in May 2003 the RIAA has sent notices to over 300 corporations claiming that their employees are using corporate Internet access and computers for uploading and downloading illegal copyrighted material [10].

By sending the notice, the RIAA has informed the company that if no actions are taken to prevent the illegal file sharing, RIAA may sue the company for passively aiding in copyright infringement. However, there are no decisions available if the employers can be held liable.

Internet service providers
According to U.S. legislation, an ISP can be held liable for contributory infringement if it has knowledge of the infringing activity and is still causing or materially contributing to the infringement conduct. However, the risk should be rather low.

In the U.S., the Digital Millennium Copyright Law gives copyright owners a right to subpoena ISPs to reveal P2P user's identity and personal information. Similar demands have also been sent to the European operators even though the national laws may prohibit the processing or delivering of the identification information for this purpose.

The subpoena is typically pressed, e.g., by a threat of loosing peering agreements. Therefore, the European ISPs may be bullied to break the privacy legislation. Also in the U.S., the smaller ISPs that do not have sufficient legal resources to evaluate the subpoena may break the privacy legislation by reacting on over broad subpoenas.

Internet service providers may also want to limit the use of P2P applications due to the huge amount of data traffic generated by the P2P applications. The limitations can be technical or just based on service contracts. However, there are two issues that the ISP has to take into account:

- When basing usage restrictions only on the service contract, the provider has to understand that the used constrains may be hard to monitor in a legal manner and that the restrictions, such as usage restrictions of P2P applications or servers, are difficult to define unambiguously.
- When using technical restrictions, the ISP has to document and present the restrictions to the customer.

P2P developers
After peer-to-peer application users, the P2P technology, application or related service developers have the greatest risk of being subpoenaed. The problem is that the developer cannot prevent copyright infringements if he wants to design an open and flexible system that can be used to share many kinds of content.

Nevertheless, after being notified about copyright infringements of the users of the system, the developer can be considered at least partly responsible for the future infringements as was the case with Napster. Therefore, the legislation and interpretation of copyright laws are strongly directing the development of peer-to-peer systems. EFF is giving guidance [12] for P2P developers to avoid the copyright infringement subpoenas.

Developers can be considered liable for both direct and indirect copyright infringements. The developer can be held liable for direct infringement if the system makes or distributes copies of copyrighted work. Therefore, also the implementation of caching and similar activities should be carefully considered.

Regarding indirect liabilities, the possibly risky features and business models are much harder to avoid. For example, according to U.S. legislation which is most likely to be used in cases against P2P developers, the developer can be found liable for both contributory and vicarious infringement that are described in more detail in Section 3.1 Federal Legislation.

3 U.S. Legislation

In the U.S. there are many federal copyright laws but only a few of them are relevant considering the scope of this paper. The main principles are same internationally and also the U.S. laws implement the international copyright treaties presented in Section 2.2 Copyright.

The federal laws are generally mirrored to the state level legislation but some differences may still occur between the states. Not only the differences, but also the interpretation of these laws cause problems, e.g., in the area of privacy and penalties. For further information on U.S. copyright legislation see [13].

The federal copyright legislation that need to be taken into account in peer-to-peer communication and system development is presented in Section 3.1 Federal Legislation and some of the burning legal issues will appear in Section 3.3 Implications and Legal Issues. A brief study of American bodies acting for or against P2P file sharing is presented in Section 3.2.

3.1 Federal Legislation

Federal level copyright legislation comprises of a handful of laws that in the U.S. govern the copyright practices. Copyright infringements can be divided into direct and indirect. Direct infringement consists of direct violation of copyright owner's rights, e.g., by sharing copyrighted material.

Indirect infringements can be divided into two court-created categories: [12]

- Contributory infringement is commonly defined such that the contributory infringer knew about the infringement and still induced, caused or materially contributed to the underlying direct infringement.
- Vicarious infringement is typically defined such that the vicarious infringer had a right and ability

to control the direct infringer and received a direct financial benefit from the infringement.

U.S. Copyright Law

U.S. Copyright Law [14] defines the basic rights of the copyright owner as well as the limitations of these rights. The law also defines the duration of copyright and penalties for copyright infringements in both civil and criminal cases.

The civil remedies can be applied to generally any copyright infringement while criminal penalties are applied in the case of intentional acts for commercial advantage, private financial gain or possibility of financial loss to the copyright holder. The maximum penalties are very high. For example, in a civil case, the violator can be held liable for damages up to $150 000 per work.

Article 107 of the law defines the fair use doctrine that allows a limited use of copyrighted material without a need to ask permission. The legal use cases include, e.g., copies for classroom use, criticism, comment or news and quotations for research. Also some other limitations exist, but any of them does not generally permit making copies for private use, e.g., by downloading shared files from P2P network.

The Digital Millennium Copyright Act

The Digital Millennium Copyright Act (DMCA) [15] implements the WIPO Copyright Treaty and WIPO Performances and Phonograms Treaty. DMCA addresses numerous issues from which the following are most relevant concerning the P2P systems.

The DMCA protects ISPs from copyright infringement liability for transmitting information over the Internet. However, it also introduces a notification mechanism between ISPs and copyright owners. When notified, the ISP has to act to remove the copyrighted works, e.g., from the user's web site or it may be held liable for any resulting damages.

The DMCA also allows copyright owners to issue subpoenas to ISPs to retrieve the identities of users and personal information merely based on good faith belief. Thus, no permission from any lawsuit is required. However, some cases are currently being questioned in U.S. courts due to privacy concerns, and therefore the limits of this right are yet controversial.

No Electronic Theft Act

No Electronic Theft Act [16] attempts to reduce digital piracy by introducing criminal penalties for copyright infringements by electronic means.

In addition, the act amends the definition of commercial advantage or private financial gain to include any possible gain such as increased participation level in Kazaa. Therefore, all free riding prevention mechanisms used in P2P file sharing applications may make user more liable.

As can be seen, the mass use of the Internet has already caused changes to legislation, e.g., by introducing the No Electronic Theft Act. Also the P2P file sharing is now affecting the legislation, e.g., according to a new proposal, even an attempt to videotape a film in the movie theatre is considered illegal.

3.2 American Groups Acting for or against P2P File Sharing

In the U.S. there are many strong lobbying groups representing the copyright owners. The main groups include: [10]

- Business Software Association (BSA)
- Recording Industry Association of America (RIAA)
- Motion Picture Association of America (MPAA)
- National Music Publishers Association (NMPA)
- International Federation of the Photographic Industry (IFPI)

In addition, some individual copyright owners, such as Universal Motion Pictures, have been trying to prevent the sharing of their copyrighted material via P2P systems. This section presents the RIAA as the most notable player against and EFF as the most notable group for P2P file sharing applications. Also P2P United is briefly introduced.

The Recording Industry Association of America

The Recording Industry Association of America (RIAA) [17] is a trade group representing most of the U.S. recording industry. One of the main goals of the RIAA is to protect its members´ copyrights globally.

For example, in 2002 the RIAA launched an anti-piracy initiative focused on sellers of pirate CDs in retail outlets, flea markets or through websites and underground communities.

In summer 2003, the RIAA started suing the P2P file sharing users. According to the announcement, the RIAA is using software to scan user's shared folders. The RIAA identifies heavy users and their ISPs. After that RIAA subpoenas the ISP to get the user's identity and personal information to be able to sue him or her to court.

The RIAA has also launched a Clean Slate Program [18] that offers amnesty to P2P file sharing application users who voluntarily identify themselves and pledge to stop illegally sharing music on the Internet.

Concerning P2P networks, the RIAA has also been suing the P2P developers and according to a recent newsletter [19] the RIAA will sue the P2P developers if the following reforms are not implemented:

- Change the default settings so that users are not automatically uploading content from their hard drives.
- Notify the users clearly that the downloading and uploading of copyrighted material without permission violates the federal law.
- Filter the protected works.

Electronic Frontier Foundation
Electronic Frontier Foundation (EFF) [20] is a U.S. based donor-supported organisation to defend the rights to speak, think and share ideas thoughts using new technologies such as the Internet.

EFF lobbies and educates press, policy makers and the public about civil liberties. EFF opposes the legislation that it thinks to be misguided by making proposals and by initiating and defending court cases preserving individual's rights. In addition, EFF, e.g., publishes papers, hosts events and keeps a comprehensive archive of digital civil liberties information at their www-pages.

Concerning P2P applications and systems, EFF has begun to meet P2P developers to discuss the possible legal challenges they may face. EFF has been giving advice how the developers can limit their legal liabilities. EFF has also announced that it is preparing to defend the developers if the need will rise. A good example of the help EFF is giving is the conference paper presented in a P2P conference [12].

P2P United
P2P United [21] is a non-profit trade association founded by five U.S. P2P developers to lobby policy makers and members of Congress. Its mission is to polish the image of P2P technologies so that the policy makers would allow responsible P2P file sharing application developers to exist.

P2P United has published their member code of conduct that introduces the reforms required by the RIAA. The P2P United announces clearly that it has nothing to do with Kazaa and it seems that the founding of P2P United is a real effort from the P2P industry to avoid the lawsuits.

3.3 Implications and Legal Issues

As presented earlier, the U.S. legislation and the RIAA are making the lives of P2P users and developers much harder. Users are under a constant threat of being sued and the P2P developers have to change their products rather dramatically to avoid being sued by the RIAA.

The use of P2P file sharing applications has already been decreased in the U.S. and the legal consequences may lead to a huge change in P2P file sharing applications.

In the Clean Slate Program [18] the RIAA promises not to support or assist copyright infringement suits based on past conduct after the user has signed the affidavit. However, the program includes also some problems that the user has to take into account before posting the affidavit:

- According to the program description, the promise applies only if the RIAA has not already begun to investigate the user in question. The problem is that the user cannot have certain knowledge if he is being investigated and therefore the user may unknowingly admit the infringement without even having a possibility to receive amnesty.
- The RIAA does not have right to grant full amnesty because it presents only 90% of all U.S. sound recording copyright owners. This may be the program's main problem because the affidavits can be used against the user in other infringement lawsuits.

4 EU Legislation

It is important to study the EU legal framework because the European Community regulations are binding and directly applicable in all EU countries. Directives are to be adapted to national legislation within a certain time frame and decisions obligate the named member state governments and private persons.

The main directives concerning peer-to-peer services and especially P2P file sharing are the Copyright Directive and Directive 2002/58/EC on privacy and electronic communications. Also the draft IPR Enforcement Directive is briefly discussed below.

The Copyright Directive
Like the U.S. Copyright Law, also the Copyright Directive (Directive 2001/29/EC on the harmonisation of certain aspects of copyright and related rights in the information society) [22] defines the basic rights of the copyright owner.

The Copyright Directive lists a number of exceptions and limitations that member states may provide concerning these rights. Contrary to U.S. legislation, the EU member states may, e.g., provide natur al persons a right to copy copyrighted works for private, non-commercial use.

The Copyright Directive implements the WIPO Copyright Treaty. Therefore; it also introduces the exceptions that enable ISPs to transmit user data legally over their networks and permit certain browsing and caching activities. The Copyright Directive mandates the member states to provide appropriate sanctions and remedies for copyright infringements.

Directive on Privacy and Electronic Communications

Directive on Privacy and Electronic Communications (2002/58/EC) [23] harmonises a level of protection of fundamental rights in electronic communications. The directive is focused on privacy and processing of personal data.

The directive permits the processing of identification information for a few specific reasons such as billing, detecting technical failures and errors and detecting and preventing fraud. Thus, the operator is not explicitly permitted to independently investigate the copyright infringements or deliver the personal data to the copyright owner.

According to the directive, the use of spyware is allowed only for legitimate purposes with the knowledge of the users concerned. This statement makes the spyware distributed by the American copyright owners illegal in Europe.

Draft IPR Enforcement Directive

The European Commission proposal for IPR Enforcement Directive [24] is proposing strict actions against IPR infringements. The aim of this directive is to harmonise the national legislation on the enforcement of intellectual property rights.

However, the proposal contains also many problems, e.g., in the areas of privacy, fair use and software competition. Concerning P2P systems the main changes come from the area of processing the identification information and from the liability issues.

According to the proposal, telecom operators and ISPs could possibly be held partly liable for infringements. More importantly, the proposal enforces the ISPs to retrieve and hand over the personal information of a suspected infringer. The U.S. copyright owners may also start ordering ISPs to carry out surveillance of their customers and to stop copyright infringements by blocking the traffic.

The proposal is aimed to cut down the copyright infringements and it could really decrease the illegal file sharing. The proposal may also have an effect on legitimate P2P usage and combined with the P2P cases in the U.S. we can claim that the emerging legislation is clearly directing the development of P2P file sharing architectures.

According to the Foundation for Information Policy Research (FIPR) report [25] the proposal has been lobbied mainly by the Hollywood and music industry. Thus, it is easy to understand the perspective of this proposal. The European operators and ISPs are strongly opposing the directive and resistance is also building, for example, in the European press. Therefore, it is still too early to say, in which direction the proposal will evolve.

5 Case Study: Finland

The Finnish Copyright Law implements the EC Copyright Directive and thus provides the same rights and restrictions. In addition to the mandatory legislation, the current Finnish copyright legislation permits a user to copy material for his own use from any source he wants.

However, the fresh version of the Government Bill (HE 177/2002) [26] for the new copyright law is proposing that the copying is permitted only from a legal source such as a library or a CD bought from a store. The parliamentary discussion is still ahead but the proposed change is still a clear indication of the effect of P2P networks.

The Finnish legislation on the processing of the identification information is discussed further in the next section.

5.1 Processing of Identification Information

Identification information comprises user's phone number, IP address or other information created and stored during the session. The processing of this information is regulated by Section 3 of the Protection of Privacy and Data Security in Telecommunications Act [27].

Section 3 of the Act defines for which purposes the operator may process identification information and to whom this information can be submitted. According to the Section operators can process identification information for certain purposes such as billing but they are not allowed to process the identification information to investigate copyright infringement cases.

According to Section 18 of the Act [27] the police is entitled to obtain, upon the consent of the injured party and the party in possession of the subscription, identification information about calls necessary for the investigation of a crime.

Such crimes are, for example, violation of protection order and disturbance of domestic peace. According to the Act copyright infringement is not considered to be severe enough that the information could be obtained. Therefore, the police cannot use the identification information in investigation of copyright infringements.

However, the Finnish Government has proposed a new act on privacy in electronic communications [28] and submitted the proposal to the Parliament on the 24th of October of this year. The Act will implement the European Union Directive concerning the processing of personal data and the protection of privacy in the electronic communications sector.

The Act is purposed to clarify the processing of identification information. The other major changes include new rules for processing and accessing location data and a right to filter out illegal marketing email and malicious programs in order to ensure communication services.

According to the proposal the processing rights and duties are also applied to the corporate and community customers that process identification information. The telecommunication operators are obliged to store the access data related to the process of identification for a period of two years.

When this act comes into force, the police would have better access to the information on holders of dynamic IP addresses also in the case of copyright infringements. However, it is still unclear if ISPs are allowed to investigate or deliver the retrieved identification information to the party claiming a copyright infringement.

Thus, it seems that the Finnish legislation still protects the privacy of the P2P user although the pressure to change the legislation in favour of copyright owners is substantial.

6 Conclusions

No especially peer-to-peer specific legislation exists. Nevertheless, all other relevant laws such as communication, copyright, competition, security and privacy legislation are applied. Especially the emerging enhancements for privacy and copyright legislations and the interpretation of copyright laws are strongly directing the development of peer-to-peer systems.

The P2P developers need to take the legal environment into account from very beginning of their development process so that they will not be held liable for copyright infringement. Accordingly, especially the developers of P2P file sharing systems have to primarily build their business model and system architecture to minimise the legal risks as the technical efficiency and superiority are just secondary targets. Therefore, we can guess that the global P2P systems will introduce filtering limiting the access.

The copyright owners have lobbied the policy makers and are now using the new anti-piracy rules to eliminate the distribution of copyrighted material in P2P networks. These actions seem to have worked since the use of peer-to-peer networks has decreased during last few months. However, the users may just be moving to smaller, closed P2P groups that are much harder or even impossible to control.

The Internet has expanded much faster than the laws have been able to respond. Now the legislatures and courts are trying to catch up to protect intellectual property rights while still maintaining the free flow of information over the Internet. Good examples of this development are the government bills in the U.S proposing to criminalise even the attempt to videotape a film in a movie theatre and in Finland proposing to limit the user's right to copy material for private use only from legal sources.

Also the right of the police to intercept P2P traffic or retrieve identification information as well as rights of network operators and ISPs to limit the traffic and process the identification information have to be adjusted to the P2P communication environment. Therefore, we can claim that also the increasing use of P2P systems imposes new requirements that need to be adapted to the legislation.

Acronyms

3GPP:	3rd Generation Partnership Project
DMCA:	Digital Millennium Copyright Act
EFF:	Electronic Frontier Foundation
ETSI:	European Telecommunications Standards Institute
GATT:	General Agreement on Tariffs and Trade
IETF:	Internet Engineering Task Force
IPR:	Intellectual Property Right
ISP:	Internet Service Provider
LI:	Lawful Interception
OS:	Operating System
P2P:	Peer-to-peer
PSTN:	Public Switched Telephone Network
RIAA:	Recording Industry Association of America

SIP:	Session Initialisation Protocol
TRIPS:	Trade Related Aspects of Intellectual Property Rights
U.S.:	United States
WCT:	WIPO Copyright Treaty
WIPO:	World Intellectual Property Organization

7 References

[1] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, A scalable content-addressable network, ACM SIGCOMM, August 2001

[2] Kazaa www-pages, http://www.kazaa.com/, reviewed 8.11.2003

[3] B. Gu and S. Järvenpää, Are Contributions to P2P Technical Forums Private or Public Goods? – An Empirical Investigation, Workshop on Economics of Peer-to-Peer Systems, Berkeley, June 2003

[4] Dotcom Scoop, Article: Internal the RIAA legal memo regarding KaZaA, MusicCity & Grockster, 25.9.2001

[5] Kazaa Media Desktop version 2.5.1, 11.7.2003

[6] TechTarget, http://whatis.techtarget.com/, reviewed 8.11.2003

[7] World Intellectual Property Organization, www-pages: http://www.wipo.int/, reviewed 7.11.2003

[8] Berne Union, Berne Convention for the Protection of Literary and Artistic Works Author, 1886

[9] World Intellectual Property Organization, WIPO Copyright Treaty (WCT), 23.12.1996

[10] AssetMetrix Research Labs, Corporate P2P (Peer-to-Peer) Usage and Risk Analysis, July 2003

[11] S. Kotilainen, Musiikkia verkosta, Tietokone extra – Digitaalinen kuva & ääni, October 2003

[12] F. von Lohmann, Peer-to-Peer File Sharing and Copyright Law: A Primer for Developers, 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03), February 2003

[13] U.S. Copyright Office, www-pages: http://www.copyright.org/, reviewed 12.11.2003

[14] U.S. Copyright Office, Circular 92: Copyright Law of the United States of America, June 2003

[15] Digital Millennium Copyright Act (H.R.2281), October 1998

[16] No Electronic Theft Act (H.R. 2265), December 1997

[17] The Recording Industry Association of America, www-pages: http://www.riaa.com/, reviewed 6.11.2003

[18] The Recording Industry Association of America, Clean Slate Program, http://www.musicunited.org/, reviewed 20.11.2003

[19] The Recording Industry Association of America, newsletter: RIAA Challenges P2P Networks To Finally Act Like Responsible Corporate Citizens, 30.9.2003

[20] Electronic Frontier Foundation, www-pages: http://www.eff.org/, reviewed 10.11.2003

[21] P2P United, www-pages: http://www.p2punited.org, reviewed 19.11.2003

[22] Directive 2001/29/EC of the European Parliament and of the Council on the harmonisation of certain aspects of copyright and related rights in the information society, Copyright Directive, 22.5.2001

[23] Directive 2002/58/EC of the European Parliament and of the Council on privacy and electronic communications, 12.6.2002

[24] European Commission, Proposal for a Directive of the European Parliament and of the Council on measures and procedures to ensure the enforcement of intellectual property rights, IPR Enforcement Directive, 30.1.2003

[25] Foundation for Information Policy Research, Analysis of draft IPR Enforcement Directive, 31.7.2003

[26] The Finnish Government Bill (HE 177/2002) for the new copyright law, reviewed 19.11.2003

[27] Protection of Privacy and Data Security in Telecommunications Act (565/1999; Finland), 22.4.1999

[28] Finnish Government, proposal for a new act on privacy in electronic communications, 24.10.2003

Part II: SPAM

The second part of this report deals with unsolicited bulk e-mail that is now flooding in the Internet and is said to form half of all e-mail traffic. Such e-mail can be commercial or non-commercial in nature. But it is always unwanted by the receiver. The amount of unwanted e-mail has grown to the level where many users have reduced using e-mail –services and for all of us, our confidence in this communications method has reduced.

This part contains three papers that describe the phenomena, analyze the mechanisms used to counter attack the flood of spam and finally will discuss the economics behind Spam.

The papers are:

Spam – from nuisance to Internet infestation

Carl Eklund
Nokia Research Center
P.O. Box 407, Fin-00045 Nokia Group
carl.eklund@nokia.com

Abstract

The phenomenon today known as spam was first seen more than 25 years ago. The recent explosion in the volume of e-mail spam may pose a direct threat to e-mail as we know it today as well as degrade the usability of the Internet. This paper gives an overview of the various types of spam and examines spam related Internet phenomena. It also presents techniques used by spammers.

1 Introduction

Most Internet users have a first hand experience of spam13. It clogs the mailboxes of many users and for those fortunate enough not to suffer from continuous floods of unwanted e-mail messages, it may degrade the service and in the worst case even render it useless. Spam is not a new phenomenon, the first incident of what today could be considered spamming occurred in 1971, and the first e-mail borne spam was conceived in 1978. However, the phenomenal growth in the volume of spam on the Internet during recent years has brought the phenomenon into the spotlight and made spam a household word.

The first known spamming incident took place in 1971on Compatible Time Sharing System in MIT. A system administrator used the mail feature in the system to send all users an anti-war message. In 1978 the first e-mail spam was sent to all users of the ARPANET on the west coast. The message was an invitation to various receptions in California where the new DEC-20 computer was promoted.

The first USENET spam saw daylight in 1988 when a college student sent out a message, with a plea for donations to his college fund to all newsgroups he could find. This posting caused at the time a lot of debate about the merit of allowing sites that sold accounts to the man on the street.

While the incidents mentioned above clearly were spam, they were not called such at the time, as the phrase spam wasn't coined until 1993. Richard Depew was advocating changes to the way USENET was moderated. He was promoting the idea of retro-moderation, i.e. moderation of a group by cancelling offensive posts after they have been submitted to the newsgroup. Unfortunately, his administration tool, called ARMM, contained a bug that lead to the posting of 200 messages in a row on the net.admin.policy group. People familiar with the Multi User Dungeon (MUD) community were quick to call his posts spam and Depew was quick to apologize for having 'done a spam'. The term spam had been used in MUDs already in the 1980s but this was the first time the word originating from the SPAM skit in Monthy Python's Flying Circus was used in the context in which it is used today [1].

2 Definitions of spam

Several conflicting definitions can be found for the word spam. In the anti-spam and ISP community spam has always been bulk and unsolicited. On the web site Monkeys.com signatures are collected to endorse the following definition of spam[], which also will be used in this paper: "*Internet spam is one or more unsolicited messages, sent or posted as a larger collection of messages, all having substantially identical content.*" The Spamhaus Project proposes a third aspect to the definition giving the following definition for spam [2]: "*An electronic message is "spam" IF: (1) the recipient's personal identity and context are irrelevant because the message is equally applicable to many other potential recipients; AND (2) the recipient has not verifiably granted deliberate, explicit, and still-revocable permission for it to be sent; AND (3) the transmission and reception of the message appears to the recipient to give a disproportionate benefit to the sender.*" This definition is obviously vaguer as it relies on the recipient's perception of the message.

An erroneous and often repeated definition states that spam is unsolicited commercial e-mail (UCE). This definition leaves out the bulk aspect which from a technology point of

[13] Spam is not to be confused with SPAM, a trademark of Hormel Foods Corporation. Many Internet users also have first hand experience of SPAM.

view is the most problematic. There are also many spams that are uncommercial. Examples are religious or political spams. Also some UCE is clearly not spam. Personal e-mail inquiring about open job positions in a company is a good example of UCE that should not be considered spam.

Recently in the US, the Direct Marketing Association (DMA) together with the Association of National Advertisers and the American Association of Advertising Agencies released guidelines for what they consider legitimate e-mail marketing practices. Conveniently they then adopted the word spam to mean any e-mail marketing message that does not follow these guidelines. Specifically these guidelines state: “All commercial e-mail (except for billing purposes) must provide consumers with a clear and conspicuous electronic option to be removed from lists for future e-mail messages from the sender.” This is in stark contradiction with the conventional definition of spam since it endorses an opt-out regime [4].

Legislative ‘anti-spam’ efforts in the US have unfortunately adopted the spam definition of the DMA and it is likely that the US will have legalized certain kinds of spam by the end of 2003. The EU directive of e-marketing is opt-in and bans spam regardless of message content [5].

2.1 What users consider spam

When asked to define spam Internet users seem to agree easily on a basic definition but the borders of the definition are fuzzier. Of American e-mail users 92% agree that spam is “unsolicited commercial e-mail from a sender they do not know or cannot identify” according to a survey by the Pew Internet and American Life Project. The survey also shows that the content of unsolicited messages also determines whether a message is considered spam. Messages containing adult content are considered spam by 92%, financial deals and investment proposals are labelled spam by 89% while product and service offers were called spam by 81% of e-mail users. Unsolicited messages from religious, political or advocacy groups (76%) as well as from non-profits or charities (65%) were not as commonly labelled spam. The prior relationship between the soliciting entity and the e-mail user strongly influences the perception. Only 32% consider unsolicited messages to be spam if they had previously done business with the sender. However, 11% consider unsolicited commercial e-mail to be spam even if they have given their explicit consent to the sender to contact them [9].

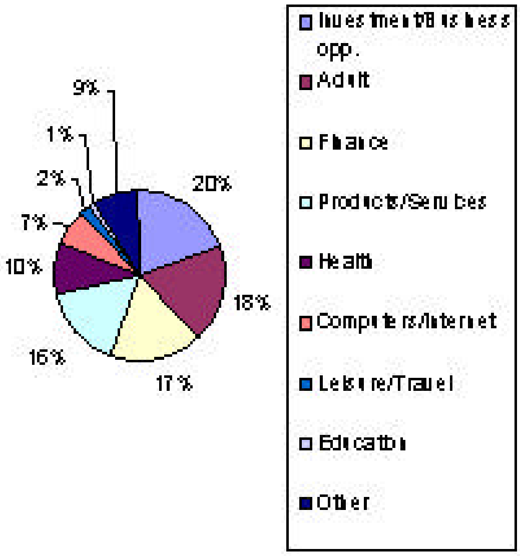
3 E-mail spam

E-mail is the most popular way of using the Internet. Of adult American Internet users 93%, about 117 million people, use e-mail. The number of e-mail messages bouncing around in the Internet any given day has been estimated to be around 30 billion. Out of these at least 15 billion messages are estimated to be spam [6]. According to anti-spam software vendor Brightmail, July 2003 saw the volume of spam exceeding the volume of legitimate e-mails. In 2001 spam was estimated to be 8% of all e-mail traffic [7]. AOL reported in April 2003 that they blocked 3.27 billion spam per *week* [8], figures from October 2003 puts the number of blocked spam at 2.4 billion per *day*. According to AOL this is roughly 80% of its incoming e-mail traffic [6].

The Spamhaus Project estimates that 90% of all spam received by users in North America and Europe can be traced back to a group of 200 professional spammers[9].

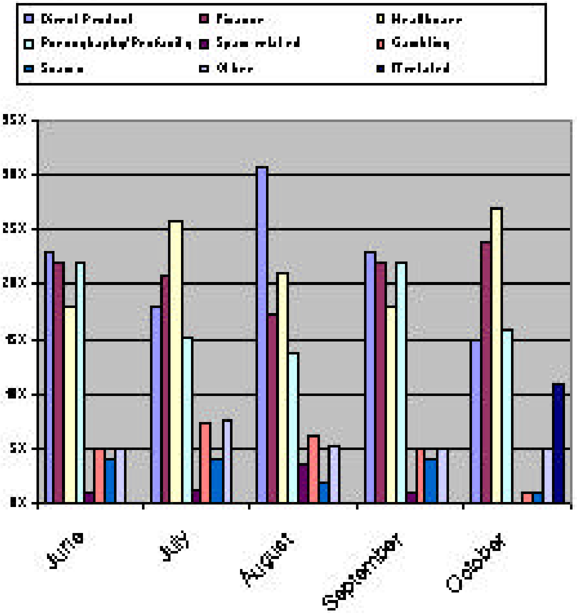
3.1 Content of spam

The Federal Trade Commission of the US studied the content of pieces of spam in April 2003. Investment/business opportunity, adult service and finance offers together made up 55% of all spam. The distribution of offers is shown in Figure 1. A third of all spam contained a false ‘From’ line. Most of these claimed to be from someone with a personal relationship with the recipient. The relationship was typically manifested by the use of a first name only. The ‘Subject’ was false in 22% of the spam. In 32% of these there was no correlation between the claimed subject and the actual message. Also claims of personal relationship were common (25%). The message itself was likely false in



Figur e 1: Distribution of offer made via spam based on randomly picked sample fr om spam accumulated up until April 2003[10].

40% of the messages with 90% of all business or investment offers falling into this category. The fraction of messages with false information in at least one field was 66% [10].



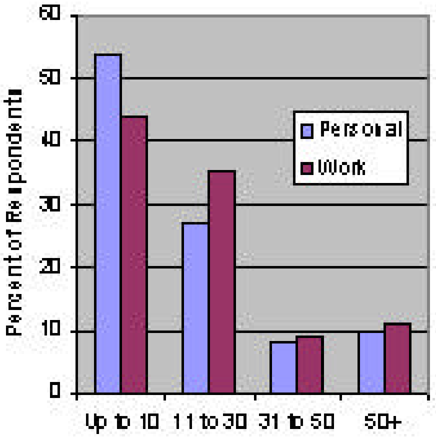
Figur e 2: Observed spam per category June-October 2003 [1 1].

It seems that spammers also are adapting the products and services marketed according to season. Figure 2 shows the prevalence of different types of spam during the period of June-October 2003. In August a large number of spam were seen promoting Arnold Schwarzenegger campaign t- shirts and J-Lo engagement rings. October saw an increase in loan offers to get consumers past the holiday season [11].

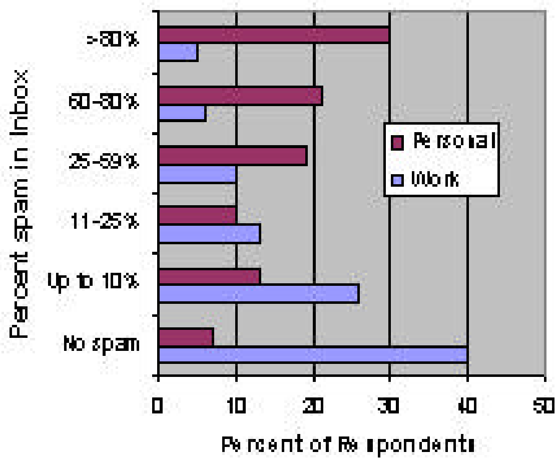
3.2 The burden of spam

Of American e-mail users about a quarter receive 5 or fewer messages per day (see Figure 3). Another quarter are heavy users that receive more than 30 messages daily with the rest of the users fairly evenly distributed in between. A third of the users found that less than 25% of their e-mail was spam, a third had more than 60% spam in their Inbox with the remaining third distributed well in between. The proportion of spam in e-mail is not strongly correlated with the amount of e-mail messages received per day. The exceptions are the users receiving the fewest e- mails who practically receive no spam and the heavy users out of whom 39% reported that more than 80% of their e- mail is spam [6]. A possible explanation for the disproportionate number of spams the heavy users get can probably be found in the active net life they live. Heavy users of e- mail are also more likely to use other Internet applications, such as USENET and IRC, and maintain web sites and blogs.

Even though e-mail users process fewer e-mail in the personal account than on their work accounts, spam is more prevalent in personal accounts (see Figure 4). Problems cited to be caused by spam are often practical and logistical in nature. Spam blocked accounts; it costs money and it takes time to deal with spam. The time spent on spam depends on factors such as connection speed, the protocol used to retrieve the messages and user sophistication. Some 40% of home users reported they spend less than 5 minutes per day on dealing with spam while 26% spend 15 minutes or more on the same task. The latter group in addition to receiving a substantial amount of spam most likely are not adept in installing and configuring spam filtering software. A majority of home e-mail users (55%) reported that spam has sometimes made it hard to get to the messages they want to read.



Figur e 3: Number of e-mails r eceived on a typical day.[6]



Figur e 4: Percent of spam in personal and work place Inboxes on a typical day[6].

Surprisingly few work e-mail accounts suffer from spam. Some 40% of people using e-mail at work report that they don't get any spam, another 26% only receives 10% spam and 12% get up to 25% spam. Almost two thirds of work e-mail users spend less than 5 minutes per day on spam. Only 10% spend more than 30 minutes on it [6].

The difference in the amount of spam between home and business e-mail accounts is the result of many factors. Businesses tend to have strict account usage policies and hence users are more careful in giving out their business e-mail address. Also businesses have better defence against spam mostly in the form of filters. Many web sites, shops and organizations ask people to provide their e-mail address. The address given is most likely the personal one. Most personal e-mail accounts are provided by a few large ISPs. These ISPs are obviously the most lucrative targets for spammers. They are also most susceptible to spam using dictionary attacks to guess receiver addresses.

3.3 User attitudes towards spam

When asked about the bothersome aspects of spam 69% percent of American users found all aspects of spam annoying.

Table 1: Aspects of spam that bothers e-mail users[6].

Bothersome aspect	% bothered
Unsolicited nature of spam	84
Deceptive or dishonest content	80
Potential damage to computer	79
Volume of spam	77
Offensive or obscene content	76
Compromise to privacy	76
Can't stop it	75
Time it take to deal with it	69

When asked to prioritize the annoying aspects, more people identified the offensive or obscene content than any other factor [6, 12]. Here the puritan cultural tradition of the US is clearly visible in the results. A similar survey conducted in Europe would most likely give a different result on this point.

Table 1: Effect of spam on e-mail use

Half of all e-mail users say that spam has lowered their confidence in e-mail in general including more than a quarter acknowledging that spam has had a big effect on their trust of e-mail. The two main agents to blame for the reduced trust in the reliability of the service are the e-mail filters deployed in the Internet and the users themselves. Some 30% fear that desired important incoming mail is being blocked by spam filters and 13% claim that this has happened to them. About 23% fear that their message will not reach the intended recipient due to spam filtering.

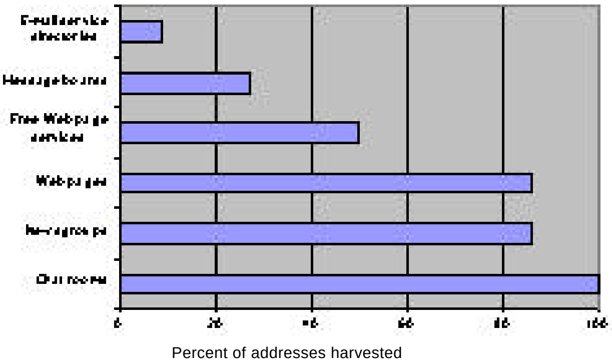
Spam has also made being on-line more complicated. Almost a third of all e-mail users are concerned about accidentally deleting an important mail message mistaking it for spam. People relying on e-mail for mission critical communications see this as a big problem as well as people that rely on e-mail for getting new business. One quarter of users say that spam has reduced their overall use of e-mail. Most of them have done this in a significant way [6].

4 Tricks of the trade

E-mail was the first real Internet killer application. When the protocol for transporting e-mail, SMTP, was developed, the Internet was very different from the Internet we know today. Parties connected to the Internet were mostly government and academic organizations pursuing non-commercial interests. The concept that Internet users are benevolent creatures that play by the rules was alive and well. This thinking is deeply reflected in the design of SMTP, the engine that allows spammers to do their business. SMTP has no provisions to authenticate the sender of an e-mail. Forging an e-mail message is trivial. Mail Transfer Agents have no way of verifying the headers that log the route a message has taken (except for the headers added by the immediate neighbour) [13]. While SMTP is the main tool of spammers, they also resort to other techniques to be able to spam more effectively. Some of these techniques are presented below.

4.1 Address harvesting

In order to spam a spammer needs a collection of e-mail addresses. Users give their e-mail addresses to web sites, shops etc. In some countries, particularly in the US buying and selling data bases containing personal information is legal, so one option a spammer has is to acquire addresses from some other party. A cheaper and more effective way of acquiring addresses is to use bots and spiders to search for addresses stored in the Internet automatically. An operation conducted by the Northeast Netforce of the FTC shows that e-mail addresses posted on the Internet are likely to receive spam (see Figure 5).



Figur e 5: Addr ess harvesting by forum[14].

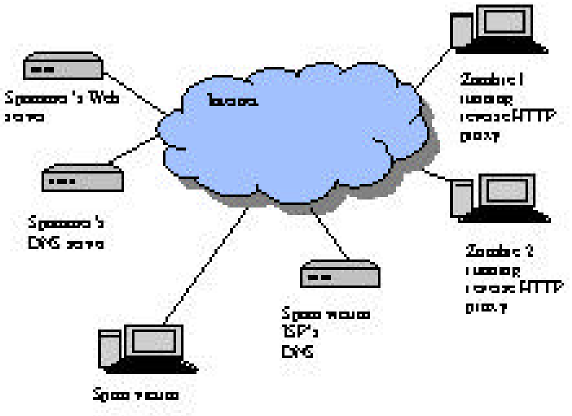
In one particular incident spam started to arrive a mere 8 minutes after the address was posted for the first time in a chat room [14].

Another method of collecting addresses is the use of methods of social engineering to get people to give their e-mail address. A company run by a notorious spammer in Florida, US, called Opt In Inc. are affiliated with web services that offer free gambling or lotteries on the condition that the user surrenders his e-mail address and consumption preferences. Fine print on a separate web page states that by taking part in these games the user also gives permission to Opt In to spam him. Opt In rents addresses for $2 per each thousand addresses to anybody willing to pay [15].

4.2 Internet zombies

Early 2003 saw the first incident of spammers, crackers and virus writers co-operating. The W32.Sobig.E, and later the F variant, trojan was designed to turn infected machines into Internet zombies, i.e. machines running open SMTP servers and HTTP proxies without the knowledge of the owner of the computer. These zombies are used by spammers to send spam and run web servers where they either sell their products and services or perpetrate their scams. It is estimated that 60% of all spam is sent via Internet zombies [16].

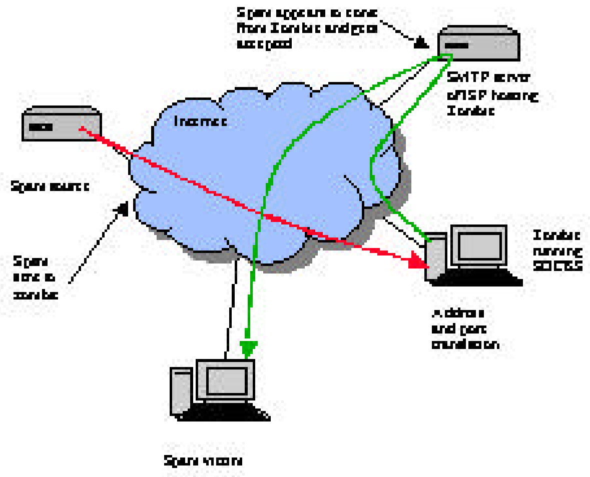
In June 2003 a spammer was observed moving his web site around seemingly at will on a minute by minute basis. What had happened was that the spammer had managed to infect thousands of systems with a small Trojan, first believed to be running a web server, called Migmaf – rotating them in and out of the DNS for the domain names he owned every 10 minutes. It made it nearly impossible for ISPs to track and shut down the server. After inspecting the code of the Trojan, it turned out that it wasn't a web server at all but instead a reverse HTTP proxy. When someone requested an URL the spammer included in his message he would be directed to an infected machine. The Trojan would forward the request to the server of the spammer and relay the response back to requester.



1. Zombies 1 and 2 get infected by trojan horse installing a revers
HTTP proxy.
2. Spam victim clicks on link in spam at t_0. At t_0 the DNS query for the host in the URL resolves to Zombie 1. The ISP DNS caches the address (lifetime e.g.10 minutes)
3. Zombie 1 retrieves the requested page from the spammers Web server and relays the page to the spam victim. Typically the spam victim becomes victim of a scam.
4. If spam victim clicks on link at t_1 after cache entry in ISP's DNS has expired. Now the DNS query resolves to Zombie 2.

Figur e 6: Internet zombies acting as r everse HTTP proxies

Additionally, the trojan runs a SOCKS proxy server on TCP port 81, allowing the spammer to bounce messages via the zombie to the intended recipient [17].


Spam appears to come from Zombie and gets accepted
SMTP server of ISP hosting Zombie
Internet
Spam source
Spam sent to zombie
Zombie running SOCKS
Address and port translation
Spam victim

Figur e 7: Bouncing spam via a zombie

4.3 Avoiding getting shut down

Spammers often portray themselves as small ISPs with throw-away domain names when buying network access. When the higher tier ISP serving the spammer, gets complaints about spam the spammer masquerading as an ISP pleads for some time to shut down their 'customers'. Eventually the spammer might get thrown out and mo ve their operation to a new ISP.

Often they also conduct business in countries where the Internet is highly unregulated, e.g. China and South East Asia are popular locations for spammer to run their operations. For some ISPs especially in poorer countries spammers that pay top dollars for access may view the spammer as exceptionally good customers.

4.4 Distributed denial of service attacks

In October 2003 several versions of a worm called Mimail were found in the Internet. This worm is programmed to perform a distributed denial of service attack on anti-spam web sites, aiming to disrupt the distribution of block lists [16],[18]. Although no hard proof exists that spammers lie behind this attack they seem to be the only party benefiting from these attacks.

4.5 Obfuscating messages to trick filters

Many mail servers today filter content on their incoming interfaces. To trick these filters spammers obfuscate the messages or insert specific components that are known to create problems for filters. A comprehensive list of tricks can be found in The Spammers' Compendium [19]. Many methods rely on random or deceiving information put inside hidden or bogus tags in HTML. Also flavours of character encoding are popular. Figure 8a-8d shows a specimen collected from the Internet.

```
PGh0bWw+DQo8YSBocmVmPSJodHRwOi8vJTc3JTc3dy5wJTYxJTcz
JTczNCU2NiU3MmUlNjUlMkVuZXQvcGIzLyIgVDhJPjxGT05UIFNJ
WkU9NT48Qj4mIzg3OyYjOTc7PCFLND50PCE0YTQ1PmMmIzEwNDs8
IVBKMHV1PiAmIzY4OzwhT1UxMGRRPm88IWgzMj5nPCFOWDc4PnM8
IUY0NzZ0PiAmIzExNTsmIzEwODs8IXkweDY+dSYjMTE0OzwhV1ZR
PnAmIzMyOzwhMW0+eTwhS1NrUD5vPCFvMzVBZT51JiMxMTA7JiMx
MDM7PCE0N2ViVTM+ICYjMTAzOyYjMTA1OyYjMTE0OyYjMTA4OyYj
MTE1OyYjMzI7PCF5MjU+cCYjMTE3OzwhOFljPnMmIzExNTsmIzEy
MTs8ITVSaTQ+JzwhcEdTNj5zJiMzMjsmIzk3OzwhQWgxPnMmIzEy
OyYjMTE2OyYjMTA0OzwhMXJKM1JIPmU8IW84V1h1PnkmIzMyOzwh
MzU+czwhMFE3ND5jJiMxMTQ7PCFSZnA+ZTwhUGw+YTwh
SzQ+bTwhNGE0NT4gJiMxMDI7PCFQSjB1dT5vJiMxMTQ7PCFPVTEw
ZFE+IDwhaDMyPm08IU5YNzg+bzwhRjQ3NnQ+ciYjMTAxOyYjMzM7
PC9mb250PjwvYT48QlI+DQo8QlIgck0wc1JhUHE+PGEgaHJlZj0i
aHR0cDovL3d3dyUyRSU3MCU2MSU3MyU3MyUzNGZyZWUlMkUlNkUl
NjV0L3BiMy8iIDFySjNSSEJvOFcgdW5TVlQ3PjxGT05UIFNJWkU9
ND48Qj48IXkweDY+QyYjMTA4OzwhV1ZRPmkmIzk5OzwhMW0+azwh
S1NrUD4gPCFvMzVBZT5IJiMxMDE7JiMxMTQ7PCE0N2ViVTM+ZTwv
Zm9udD48L2E+PEJSPjxCUj48QlI+PEJSPjxCUj48QlI+PEJSPiYj
MTM7JiMxMDsmIzY5OyYjMTA5OyYjOTc7JiMxMDU7PCF5MjU+bCYj
MzI7PCE4WWM+QiYjOTc7JiMxMDA7PCE1Umk0Pj88QlIgUlIg
Mk1PZHZjTT4NCm5vIG1vcmUgPGEgaHJlZj0iaHR0cDovL3JlbW92
ZSUyRSU2RGUlNzMlNzNhJTY3JTY1bSU2NW4lNkYlNzcuJTZFZXQv
IiBSZnBOUD5DbGljayBIZXJlPC9hPjxCUj4NCjxCUj48L2h0bWw+
DQoNCmFQcTgyTU9kICAgICAgICAgICAgICAgICAgICAgICAgICAg
ICAgICBjTUo=
```

Figur e 8a: Mesage befor e removin base64 encoding.

```
<html> <a href="http://
%77%77w.p%61%73%734%66%72e%65%2Enet/pb3/"T8I><FONT
SIZE=5><B>&#87;&#97;<!K4>t<!4a45>c&#104;<!PJ0uu>
&#68;<!OU10dQ>o<!h32>g<!NX78>s<!F476t>
&#115;&#108;<!y0x6>u&#114;<!WVQ>p&#32;<!1m>y<!KSkP>o
<!o35Ae>u&#110;&#103;<!47ebU3>&#103;&#105;&#114;&#10
8;&#115;&#32;<!y25>p&#117;<!8Yc>s&#115;&#121;<!5Ri4>
'<!pGS6>s&#32;&#97;<!Ah1>s&#32;&#116;&#104;<!1rJ3RH>
e<!o8WXu>y&#32;<!35>s<!0Q74>c&#114;<!Rfp>e<!Pl>a<!K4
>m<!4a45> &#102;<!PJ0uu>o&#114;<!OU10dQ>
<!h32>m<!NX78>o<!F476t>r&#101;&#33;</font></a><BR>
<BR rM0sRaPq><ahref="http://
www%2E%70%61%73%73%34free%2E%6E%65t/pb3/"1rJ3RHBo8W
unSVT7><FONT
SIZE=4><B><!y0x6>C&#108;<!WVQ>i&#99;<!1m>k<!KSkP
> <!o35Ae>H&#101;&#114;<!47ebU3>e</font></
a><BR><BR><BR><BR><BR><BR><BR>&#13;&#10;&#69;&#109;&
#97;&#105;<!y25>l&#32;<!8Yc>B&#97;&#100;<!5Ri4>?
<BR RR 2MOdvcM> no more <ahref="http://re
move%2E%6De%73%73a%67%65m%65n%6F%77.%6Eet/" RfpNP
>Click Here</a><BR> <BR></html>
aPq82MOd                                cMJ
```

Figur e 8b: Odd looking HTML.

```
<html>
<a href="http://www.pass4free.net/pb3/"><FONT SIZE=5>
<B>&#87;&#97;tc&#104; &#68;ogs
&#115;&#108;u&#114;p&#32;you
&#110;&#103;
&#103;&#105;&#114;&#108;&#115;&#32;p&#117;s&#115;
&#121;'s&#32;&#97;s&#32;&#116;&#104;ey&#32;sc&#114;ea
m &#102;o
&#114; mor&#101;&#33;</font></a><BR> <BR><a
 href="http://www.pass4free.net/pb3/"><FONT
SIZE=4><B>C&#108;
i&#99; k H&#101;&#114; e</font></
a><BR><BR><BR><BR><BR><BR><BR>
&#13;&#10;&#69;&#109;&#97;&#105; l&#32;
B&#97;&#100;?<BR>
no more <a href="http://remove.messagemenow.net/
">Click Here
</a><BR> <BR></html>
aPq82MOd                                cMJ
```

Figur e 8c: Same HTML after some pr ocessing.

```
<html> <a href="http://www.pass4free.net/pb3/"><FONT
SIZE=5><B>
Watch dogs slurp young girls pussyís as they scream f
more!</font></a>
<BR> <BR><a href="http://www.pass4free.net/pb3/"><FON
SIZE=4><B>Cl i c k Her e</font>
</a><BR><BR><BR><BR><BR><BR><BR>  Email Bad?<BR> no
more
<a href="http://remove.messagemenow.net/">Click Here<
a>
<BR> <BR></html> aPq82MOd
```

Figur e 8d: Message revealed.

5 Spam other than e-mail

5.1 USENET spam

USENET spamming refers to the practice of sending the same message to a large number of newsgroups. The two most famous incidents of USENET spamming happened in 1994. The first spam was the ‘Global Alert for All: Jesus is Coming Soon’ which was soon followed by the infamous ‘Green Card Lottery - Final One?’ posting. Both these spams did not use the crosspost feature of USENET but sent individual messages to each newsgroup [20].

Today USENET spam is a minor problem thanks to widespread use of automatic retro-moderation and filtering in news servers. The most widely deployed anti-spam tool is called NoCem [21].

5.2 Blog spam

A blog, also known as a weblog, is a page where a weblogger collects other webpages he/she finds interesting and keeps some form of a diary. Blog spammers use programs called bots to send spam as comments to stories posted on the blogs. The messages typically include key phrases like ‘buy viagra’ together with a link to a spammers site. A more subtle spam might show up as an innocent message but contain the URL of the spammers web site embedded in hidden HTML tags.

The aim of blog spamming is to give the impression that the web site of the spammer is tremendously popular as it is referred to in numerous blogs and thus fool a search engine, like Google, to make the web site of the spammer rank higher in the search results.

The most affected blogs are those created with the tool Movable Type. Recently plug-in modules that automatically enforce black lists and prevent relatively effectively blog spam have been introduced. Also as search engine vendors have become aware of the blog spamming it is likely that they will act to render it useless.

5.3 Windows Messenger Service spam

The Windows Messenger Service is a utility that ships with Windows NT, Windows 2000 and XP. It is meant as a way for system administrators to inform users about events or problems affecting the network in real time. It can also be used by applications and devices, e.g. a printer could notify the user about a completed print jobs. The Windows user sees these messages in a window that pops up on the screen. The Messenger Service feature in enabled by default in Windows.

The protocol used is designed such that it is close to impossible to retrace a message back to the sender. This feature obviously makes it attractive to spammers. Until recently a company in California, D Squared Solutions LLC, was offering to sell software that could send 135000 messages per hour along with a database of 2 Billion unique addresses. The same company also engaged in the practice of repeatedly bombarding users with messages offering software, at a cost of $25-30, for protecting them against the pop-up messages. The company was recently issued a restraining order at the request of the Federal Trade Commission [22].

Messenger Service spam is easily prevented by turning off the feature in Windows. Blocking all ports used by Messenger is not a viable solution as this disrupts several other applications.

On October 24th AOL started disabling the Messenger Service on the computers of their customers without any notification. It is the first time an ISP publicly admits to changing settings on customers´ computers without asking for prior authorization [23].

Subsequently, Microsoft has announced that the feature will be turned off by default in Service Pack 2 for Windows XP.

6 Spam in mobile networks

6.1 SMS spam

SMS spam typically is a message that asks the subscriber to call a premium rate number. In the UK the Independent Committee for the Supervision of Standards of Telephone Information Services (ICSTIS), the premium rate services watchdog, received 4000 complaints during the eight first months of 2003. During a typical year the number of complaints they receive is around 10000.

In a survey 63% of the people interviewed had been annoyed by SMS spam. The mobile phone operator Vodafone is currently trialing a system in which mobile phone users can easily report spam to the operator and to the ICSTIS.
The practice of sending unsolicited SMS is outlawed in many countries and due to the regulated nature of telecom networks enforcement of anti-spam rules is much easier than on the Internet [24].

6.2 E-mail spam in i-mode

NTT Docomo has been grappling with spam in their mobile network offering i-mode services for several years. The number of users of i-mode is close to 40 million. All these users have e-mail addresses in the same domain. Initially NTT Docomo assigned the users addresses with their telephone number as the username. The spammers quickly developed programs that generated random user names (11 digit random numbers) and sent spam to these users. In response to this NTT Docomo urged users to change their usernames to any alphanumerical string they wished. The spammers adapted their dictionary attacks to include also characters.

The problem in the network was not only that users received spam. Since only a small part of the number space is valid and random dictionary attacks are imprecise, the number of bounced messages became huge. In October 2001 during a single day the NTT Docomo network delivered 150 million messages (including spam) and bounced 800 million messages due to them being addressed to non-existing users.

Legislative measures and filtering together with requiring i-mode users to change addresses has since reduced the amount of spam in the NTT Docomo network but still they estimate that the percentage of spam in NTT Docomo's network is higher than in the Internet. Spam is also a financial burden to i-mode users as they pay for each received e-mail. NTT Docomo had to introduce a service to allow i-mode users to get refunds for received spam [25].

7 Conclusions

For many people spam is part of their everyday Internet experience. The rapid growth in the volume of spam has put strain on both the Internet infrastructure and users. The high volume of spam makes the SMTP infrastructure more vulnerable to other threats such as e-mail worms. A large majority of e-mail users would like to see spam eradicated, be it either by technical or legislative means. The design of the Internet, its global and largely unregulated nature together with differences in legislation across geographical borders are likely to assure that the problem of spam will get much worse before it gets better. It is likely that in order to get rid of spam, e-mail has to migrate from SMTP to some other infrastructure that would change the economical landscape for spamming radically. The obvious challenge is to achieve this without major disruptions and without sacrificing the ease of use of e- mail.

8 References

[1] B. Templeton, "Origin of the term 'spam' to mean net abuse," http://www.templetons.com/brad/spamterm.html

[2] http://www.monkeys.com/spam-defined/

[3] http://www.spamhaus.org/definition.html

[4] http://www.ana.net/govt/what/10_14_03.cfm

[5] http://www.washingtonpost.com/wp-dyn/articles/A3243-2003Oct22.html

[6] D. Fellows,"Spam–How It Is Hurting Email and Degrading Life on the Internet,"Pew Internet & American Life Project, October 22, 2003.

[7] http://www.brightmail.com/pressreleases/082003_50-percent-spam.html

[8] Official Transcript Proceeding, Federal Trade Commission, Matter no. P024407,"Spam Project", April 30, 2003,p.39.

[9] http://www.spamhaus.org/rokso/index.lasso

[10] Division of Marketing Practices,"False Claims in spam," Federal Trade Commission Report, April 2003.

[11] http://www.clearswift.com/news/pressreleases/default.asp, Monthly Spam Categorisation Breakdowns, June-October,2003.

[12] Harris Interactive, "Large Majority of Those Online Wants Spamming Ban", January, 2003, http://www.harrisinteractive.com/harris_poll/index.asp?PID=348 .

[13] Postel, J., "Simple Mail Transfer Protocol",IETF RFC-821, August 1982.

[14] Federal Trade Commission, Press release November 13, 2002, "Spam Harvest Results Reap Help for Consumers Trying To Avoid Spam," http://www.ftc.gov/opa/2002/11/netforce.htm .

[15] http://www.optininc.com/

[16] http://www.spamhaus.org/cyberattacks/index.html

[17] http://www.lurhq.com/migmaf.html

[18] http://www.f-secure.fi/v-descs/mimail_c.shtml

[19] J. Graham-Cumming, "The Spammers' Compendium," MIT Spam Conference, January 2003, http://www.jgc.org/tsc/

[20] R. Horton, M. Adams, "Standard for Interchange of USENET Messages,"IETF RFC-1036, December 1987.

[21] http://www.cm.org

[22] Federal Trade Commission, Press release November 6, 2003, "FTC Obtains Order Barring Pop-up Spam Scam, Urges Consumers to Take Steps to Protect Themselves," FTC File No. 032-3223 (http://www.ftc.gov/opa/2003/11/dsquared.htm.)

[23] Associated Press, October 24, 2003, "AOL Quietly combats Pop-up Spam Messages," (http://www.eweek.com/article2/0,4149,1362767,00.asp.)

[24] Story from BBC news, http://news.bbc.co.uk/go/pr/fr/-/2/hi/technology/3181959.stm, August 26, 2003.

[25] Official Transcript Proceeding, Federal Trade Commission, Matter no. P024407,"Spam Project", May 1, 2003, pp.277-283.

Mechanisms for Detection and Prevention of Email Spamming

Vladimir Mijatovic
Nokia Networks
P.O. Box 321, 00045 Nokia Group, Finland
Email: Vladimir.Mijatovic@nokia.com; tel:+358-50-482-07-09

Abstract

The purpose of this paper is to describe and identify spammer's techniques that they are using to evade filters, and to describe the various filtering techniques that are used in today's state-of-the-art anti-spam software. Attention is focused on the Bayesian filtering technique, as this is the most popular "intelligent" mail filtering technique today. But there are also some other methods, used in commercial or freeware software that will be mentioned. The paper does not aim to propose any particular product or solution nor the products mentioned are the only anti-spam software available on the market.

1 Introduction

The growth of spam in the last couple of years is enormous. The companies and ordinary users are now referring to spam as No1 IT problem [1], even higher than viruses or security measures. It is not strange that the number of startup companies that are solely making antispam products has tripled in 2003.

2 What is spam?

There are many definitions of spam. In fact, the big problem for filtering spam is that it is hard to define it precisely, and endless debates about this topic can be found on the Internet.

One of the better-known definitions of spam from Mail Abuse Prevention System [2] is:
An electronic message is "spam" IF:

1. *the recipient's personal identity and context are irrelevant because the message is equally applicable to many other potential recipients; AND*
2. *the recipient has not verifiably granted deliberate, explicit, and still-revocable permission for it to be sent; AND*
3. *the transmission and reception of the message appears to the recipient to give a disproportional benefit to the sender.*

This definition of spam allows the end-user to identify spam upon reception (step 3). But, as there are different people, there are different opinions, and what could be a spam for one person is not for another. And that is a problem.

The problem implies that different people will consider the same message in different ways. What may be spam for you is not necessarily spam for me, and so on.

2.1 Growth of spam

Growth of spam has been enormous. Some sources have found that growth in the last 12 months has been around 18% per month [3], some are saying less. But all the sources agree that spam is growing, and is growing fast.

For example, MSN and AOL block around 2,5 millions spam mails every day. Percentage of spam mail in all the email communications is between 50-60% [4]. AOL also said that 70%-80% of all incoming Internet e-mail traffic is blocked as spam [5].

If nothing is done, and the growth stays as it is today, it is estimated that there will be up to 10 000 spam messages per inbox per day in 2008.

Another problem that helps the growth of spam is that legally it is still "a gray area". It is mostly not illegal and even penalties are very low when a spammer is convicted. Even some marketers are not aware of the social or legal costs of their actions. The only effect they see is the increase in their sales. Increase in sales is good enough reason for sending spam, although it is perhaps not ethical.

2.2 Typical spammer's business model

Why spammers are sending spam? The simple answer is that they send it in order to sell products or services. You may ask the question "who is buying anything from a spammer?" The answer is: some people do buy!

The cost of sending to the spammer is minimal. That is why the spammer's business model can survive. One notorious spammer said that the cost of sending spam is $22 000 to his entire database of 250 millions email addresses. And that makes it 0.009 cents per one email! The cost of recipients is not, however. The average cost of spam per one employee is around $750 per year (if the calculation includes the mandays for deleting spam, plus CPU and storage for spam messages, additional personnel needed etc). [6]

2.3 Spammer's techniques

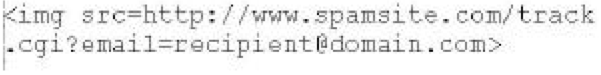
Spammers are using several techniques to send spam, hide their identity and to pass spam filters at the recipient's email systems. It is important to understand some of the commonly used ones and not only to understand the problematics of email filtering but also to realize that spammers are not naïve. In fact, spammers will use every possible technique to evade filters.

Changing Headers

A usual trick that spammers use is to forge the headers in the email. In this way, spammers try to disguise the origin of the email. Usually the `Received:` header is forged, making it hard (but not impossible) to identify the source of the email. The `From:` header is almost always forged as well.

Inserting a Picture in HTML Email

When a reference to a picture is inserted in an HTML email body, an email client will automatically fetch the picture from the embedded URL. Furthermore, if the picture is uniquely labeled, simply opening an email can indicate to the spammer that the email has been opened. This verifies not only that the recipients' email address is valid; it indicates also that the recipient has opened the spam.



```
<img src=http://www.spamsite.com/track
.cgi?email=recipient@domain.com>
```

Figure1: An example img tag that triggers a cgi script that tracks who opened the email

Another benefit of inserting a picture is that there are no incriminating words in the mail that a filter can find – all the text is *embedded in* the picture.

Invisible Text

To confuse filters, spammers are inserting text in the email that is invisible when rendered – that is, the text is white on the white background. The words in the invisible text are ideally selected to be common words that are found in average email correspondence.

An alternative would be to put words in the x-header of the email. Some spam filter software programs are inspecting the words in x-headers. The aim of putting "innocent" text in X-headers is solely to trick the filters to bypass the spam.

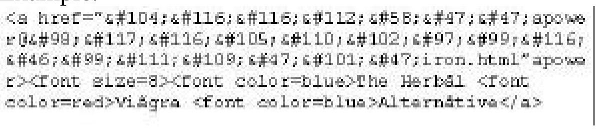
Bogus HTML Tags With a Large Amount of Text

Other option would be to put the text in an invalid HTML tag. The invalid tags will not be displayed in an email client (invalid tags are ignored when HTML is rendered) but the text may confuse filters.

Hiding a URL by various encoding techniques

There are many ways how a URL can be encoded. To hide the URL spammers are putting different encodings for URL in the `href` tag. An email client that can render HTML will decode it easily.

Example:



```
<a href="&#104;&#116;&#116;&#112;&#58;&#47;&#47;apowe
r@&#98;&#117;&#116;&#105;&#110;&#102;&#97;&#99;&#116;
&#46;&#99;&#111;&#109;&#47;&#101;&#47;iron.html"apowe
r><font size=8><font color=blue>The Herbál <font
color=red>Viágra <font color=blue>Alternátive</a>
```

Figure 2: href tag from spam mail.

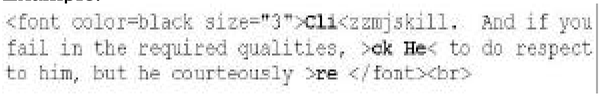
Use of JavaScript in email

Sometimes the whole message body is encoded in JavaScript. The displayed message is created when the JavaScript is executed inside the email client.

HTML tags That Are Breaking Words

If the invalid tags are inserted on purpose, the words that would normally be recognized by a filter are broken into parts and are not recognized.

Example:



```
<font color=black size="3">Cli<zzmjskill.  And if you
fail in the required qualities, >ck He< to do respect
to him, but he courteously >re </font><br>
```

Figure 3: An Example HTML from a spam mail. Bogus tags are inserted to confuse the filter

Notice in the example above how several tricks are used by the spammer. A large amount of text is inserted into bogus HTML tags. At the same time, bogus tags are separating the words that will be visible after the email reader renders the HTML.

Insert text into table

One variation to confuse the filter is to put the text into vertical tables. The text will be in several one-column vertical tables but the rendered text will be readable by a human normally.

Example:

Bu	▨	y Via	▨	gra

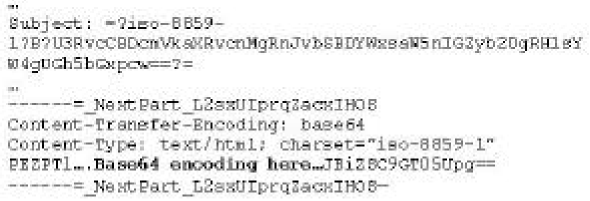
call		to		day
0800		1111		111

In the example above, each white column (with text) is one vertical table in HTML. The gray columns are either empty tables or non-existing. The rendered HTML would not show the gray columns (shown above just for clarity).

Different encoding

Some email filters are not parsing the email message body or headers in the same way that email clients do. By encoding e.g. the message body in base64 encoding the content of the body is unrecognizable to filters if the filters are not decoding base64.

Example:

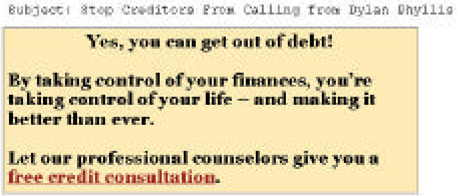


```
…
Subject: =?iso-8859-
1?B?U3RvcCBDcmVkaXRvcnMgRnJvbSBDYWxsaW5nIGZybZ0gRHlsY
W4gUGh5bGxpcw==?=
…
------=_NextPart_L2sxUIprqZackIHO8
Content-Transfer-Encoding: base64
Content-Type: text/html; charset="iso-8859-1"
PEZPTl….Base64 encoding here…JBiZ8C9GT05Upg==
------=_NextPart_L2sxUIprqZackIHO8-
```

Figure 4: An example spam mail encoded in base64. Some headers and most base64-encoded body removed from example

Notice the spam mail example shown above. The header has been encoded in iso-8859-1 and the body in base64. If the spam filter would not parse the email, the message would pass to inbox.

The example above looks in an email client like this:



Subject: Stop Creditors From Calling from Dylan Phyllis

Yes, you can get out of debt!

By taking control of your finances, you're taking control of your life – and making it better than ever.

Let our professional counselors give you a free credit consultation.

Figure 5: Decoded spam mail from the previous example

Change Letters With Accented Letters or Numbers

This is one of the most used techniques. In order to prevent a filter from "seeing" the incriminating word, the word will be changed like this:

- WORD -> W0RD ("zero" instead of "O"),
- PHRASE -> PHR4SE (4 instead of A),
- SIMILAR -> S1M1LAR ("1" instead of "I").

Or there will be a a non-English language letter that humans will ignore when reading thext:

- WORD -> WÖRD; PHRASE -> PHRÄSE, …

This trick is likely to confuse simple filters that are looking for exact matches of certain words.

Multipart MIME body

If an email is MIME/multipart, the body contains two parts [6]. One part is usually `text/html` and the second part is `text/plain`. The idea behind this is that the mail reader will display the HTML part if it is capable of doing so but if it can't, it will display the plaintext.

Most of today's email readers are rendering and displaying HTML messages. Therefore the spammer can insert arbitrary text in the `text/plain` part to trick filters.

Change Words That Are Still Humanly Readabl e

This is the usual spammer's technique. Spaces, or some other characters separate the incriminating word. An example would be **W O R D, or W_O_R_D or W'O'R'D.** This will confuse simple mail filters that have "WORD" in their simple filtering criteria.

Adding Random Words or Random Text at the End of the Message

Very often, at the end of the message, a long text of random word is attached. It is displayed at the rock bottom of the message usually in the smallest font. The purpose of the text is to confuse the filters to let the message pass. Another purpose is to increase the word count in statistics filters' tables making them unusable in the long term.

Still another purpose is to change the signature of the email. If the random text or symbols are different for each mail to be sent (and some spammers' software can actually do that), the signature of the email will be different for each recipient. This will slow the sending of the messages but just slightly (because the sender has to add a different random string at the end of each message and that requires minimal additional processing time) but will make signature-based filtering useless.

Changing Sender's Email Address Header

Sender's email address is almost always changed. The design of the email protocol is such that it allows anyone to put arbitrary text in the `Sender:` field.

Spammers are forging the `Sender:` field by putting usually the recipient's email (or some modification) in the `Sender:` address, thus trying to confuse the filters.

Putting Username in Email Message

Often one can find his username in the message body. As the username is often the name of the recipient, it is very likely that there are rules in the filter that would allow the mail to pass if there is user's name in it.

Spammer' software will take the recipient address, e.g. `john.smith@hut.fi` and will put in the body of the email something like *"Dear John Smith"*, *"Dear john.smith"* or similar.

Putting Pictures Instead of Text at Some Parts

One trick that spammers may use is to replace some parts of a word with the embedded .jpg picture of the letter. The rendered text will look like the normal text while some letters or tokens of letters will be, in fact, embedded pictures.

3 Spam prevention techniques

This Section explains today's spam prevention techniques. Spam prevention techniques can be divided into 3 wide areas: *anti-spam laws, filtering techniques (on ISP/server side and on client/consumer side)* and *other methods.*

3.1 Anti-spam laws

Anti-spam laws are a reality. There are many of them but their effectiveness is questionable. There has been lots of debate about them and some sources claim that efficient anti-spam laws could reduce spam by 80%. [7]

But there are problems in today's laws – there are holes in the law inserted there on purpose by lobbyists and direct marketing associations. Furthermore, any law to limit email sending can be misused against the basic principle of email – freedom of speech, anonymity and possibility to send unsolicited email to anyone.

In order for a law to be effective, there is a need to define spam in such a way that an ordinary mail can't be mistakenly referred to as spam. This is a task that is almost next to impossible to achieve. Another problem is that there is no body that can enforce the law effectively because the source of spam may be in another country.

When an Internet user subscribes or registers her email address to any web site, the site's policy has to be read carefully. Some of the services just sell the email addresses; others are entering into"partnerships" with direct marketers (read: spammers) and that partnership allows sending of email based on the current laws.

3.2 Corporate Requirements

Corporate requirements for anti-spam are much different from the requirements of individual users. While an individual user is mainly annoyed with spam and would like to receive as little spam as possible, the same can't be directly applied for the corporations.

Corporations have much less tolerance to false positives because email is one of the essential business tools. Failing to receive important business email may have a negative business impact. On the other side the reduced productivity because of spam that employees are receiving can cost large corporations millions per year.

One important aspect that is usually not mentioned is that an employer's obligation is to prevent "hostile work environment". An employer that has been notified by employees that they are subject of hostile emails can be indirectly liable if it does not take reasonable steps to prevent it. Note that before the employee notifies the employer, the employer cannot be held responsible. But immediately after the notification, the employer must take steps to prevent it [8].

Corporations are generally having strict requirements towards antispam products – one of the most important ones is the ability of the product to classify emails in several categories and not only spam/nonspam.

Different types of spam emails have a different effect on corporation's liability. Sexual, racial or religiously offensive material must be stopped before it reaches an employee's inbox while "special offers" may be allowed to pass (or at least have lower threshold). There are several products on the market that meet such corporate requirements; one of those is, for example, the Nokia Message Protector [9] that uses Postini technology [10].

3.3 Fighting spam on the ISP side

ISPs or email server owners are having increased costs because of spam. However, the cost of fighting spam may be even higher than the cost incurred by the spam itself because anti-spam measures require installing and maintaining anti-spam filters, enforcing email policies, blacklistng known spam sites or IPs, etc. It is a very sensitive area to an ISP as well as it is the consumers who define what is spam to them and what is not. In order for filtering to be effective, ISPs must set up individual filters, preferences, blacklists and whitelists and maintain individual filtering rules. It is not only that this process is time-consuming, it also may imply the infringement of privacy [11] as user's preferences and email behavior is maintained on the ISP side.

Also, consumer's reaction to false positives on the ISP side may create very negative publicity as happened to several providers that have filtered out important emails to their subscribers [12].

Next we explain some methods that are proposed or are in use today to fight spam on the server side.

Cost per Sent Email

A proposed solution to reduce spam is simple – if mail sending will cost for example €0.01 per email, this will not be a significant burden to the legitimate email users. This will, however, put a significant cost on spammers and their business model would not be viable anymore. Usually the expected response rate for a spam campaign is very low (less than 0.01%), hence spam would become a non-viable marketing channel for the spammers.

This proposal, simple as it may look at the first glance, has several serious flaws. The first flaw is that email is free today. To change the end-user perception of email as a free channel is not trivial and most ISPs are not willing to do so. Another problem is that by introducing a price for email, the ISP is charging the subscribers per email, however small it may be. That means that mail servers will become servers that handle money transactions. And this means that security has to be increased drastically. The potential damage to the ISP or the corporate customers will not make anyone willing to implement this approach first. Moreover, it is no use to be early adopters in this as the true benefit will be visible only then when everybody is using it.

Realtime Blackhole Lists

The idea behind the RBLs is: let's assume that there are ISPs that are known to originate spam or the ISPs and hosting providers are not taking substantial actions against the reported spammers. RBL owners will put the ISP that is reported to originate spam and that is the IP address (or range of IP addresses) of that ISP into their Lists.

Other ISPs that wish to use the RBL can download the RBL from the RBL owner or query the RBL when new mail comes to a mail server. If the mail is coming from the IP address that is on the RBL, the mail is rejected.

An ISP is removed from RBL when it stops spam originating from its domain and takes the steps to prevent spamming. Also, all open relays that are usually used by spammers to send tons of emails are automatically put into the RBL. When the open relay is closed, its IP address may be removed from the RBL.

A good side of this approach is that lot of ISPs are using RBLs and they are a fairly effective weapon in forcing all the other ISPs to prevent spam originating from their domains. If they fail to do so, lots of mails from their legitimate users will be blocked and perhaps not only email but also all IP traffic as well. This gives a real force to the hands of anti-spammers in fighting against spam.

This approach has several serious drawbacks, however. The RBLs are usually managed by a small number of individuals. The initiative is usually non-commercial and is "for every Internet citizens' good". The problem is that the list owners can blacklist someone's IP address range by mistake, due to a false spam report or just for revenge. This also causes a lot of problems to legitimate users.

For example, one of the well-known RBL owners is having a very tough policy towards any company that uses emails for notifying subscribers. They are basically giving the ultimatum to the company to implement a verified-positive email challenge-response before someone is added to the mailing list. If the company does not want to do so, its IP range is blacklisted.

Julian Haight, the owner and administrator of SpamCop (one well-known RBL) says [13] "We list you immediately, and then we can talk about it . . . I look at it as what we need to do to effectively filter out the spam. If you're innocent until proven guilty it's not an effective (way) to filter out the spam."

Many see this as a too radical approach. Even the biggest ISPs are blacklisted because of only one complaint. There is no process of how to get on and to get off the blacklist, which means that there is lots and lots of collateral damage.

Challenge-Response model

Whenever a recipient's mail server receives an email from an unknown sender, the server will reply to the sender requiring him to do something to verify that this is the real sender and not a spammer. This can be a verification on the web page, verification by sending a confirmation email or similar.

While this approach works, there are some serious disadvantages. Firstly, a sender can't send an anonymous email, which is one of the very useful features of email systems. The question is whether Internet users want to sacrifice a useful feature like this?

Secondly, the sender may not be willing to go through the tedious process just to send email. It simply requires lots of work. Some users do not understand what they should do (reply message may come in a foreign language also). Also, imagine this situation - if the recipient has set a forwarding address, the reply would come to the originating server from a different email address and in this case the server would ask the person that replied to the original mail to verify itself. This procedure is tedious and it is not hard to imagine that lots of people wouldn't want to go through so much hassle just to reply to email.

3.4 Fighting spam on the client (consumer) side

Fighting spam on the client side is the most effective approach. It allows a user to define her own set of rules and to filter out whatever she likes. There are no legal problems whatsoever as the users can choose to do with their own email what they like.

Unfortunately, most of the users find email filtering too complicated and the procedure to set up and maintain email filters too confusing, complicated and time consuming. The majority of consumers do not understand the spam mechanisms nor are they willing to invest a substantial amount of time to deal with it. 48% believes that it is enough to unsubscribe from the mailing lists to stop spam [14]. This information alone is indicative to explain the general level of user education concerning spam.

Next we will discuss some of the common methods that are used on the client side to filter spam mail. Some of the methods are widely used and some are just proposed alternatives or complementary measures. The whole area of statistical filtering is explained in the next Section, as statistical filtering is today the most sophisticated method for spam prevention and requires special attention.

Client-defined Blacklists

Most of email clients and some web-based email services, such as Hotmail, are offering blacklists. If the sender of the message appears on blacklist, the message is deleted. However, the design of SMTP protocol allows putting an arbitrary sender address in message headers. Hence, spammers trivially avoid this basic filtering.

Whitelists

Whitelists are useful in cases where some other form of filtering is applied. If the sender is on the whitelist, the email will pass to the recipient's inbox always and no filtering will be applied. Some commercial products even maintain automatic whitelists – whenever user sends email to a particular email address that email address is automatically added to the whitelist.

This approach does improve the reliability of the spam filtering. But this should be used very carefully, to avoid a situation when a simple automatic out-of-office reply to incoming spam may put that spammer to the whitelist.

HashCash

First, let's go back and review the spammer's business model. The spammer is sending millions of messages to millions of subscribers per hour. For the business model to be viable, there is no need to have a big response rate. Even one to two replies in every hundred thousand is enough because the sending is so cheap that the spammer's costs are minimal.

The approach proposed in e.g. HashCash [15] is to raise the spammer costs by requiring a process that will slow down the rate of outgoing emails from e.g. one million per hour to one thousand per hour. If this is achieved, the spammers´ business model will be ruined, as there is simply not enough spam volume to maintain a viable business.

The tactics used here is to attach to the outgoing mail an arbitrary header that will contain a field which is hard to compute but easy to verify. That will make the sender's PC compute several seconds for each email while the recipient's PC can verify the email in milliseconds.

This works as follows:

A hashing algorithm is used to produce the hash of the given plaintext. Plaintext can be of arbitrary length but the hash is of fixed length. It is, however, extremely hard to find two different plaintexts that will give the same hash. This is called collision. (Two most popular hashing algorithms today are MD5 and SHA-1). Partial hash collision is easier to find. This requires finding of two different plaintexts that have the same *n* bits of hash, where *n<TotalHashLength*. The higher the *n* the more it takes for the computer to find the plaintext that will have *n* bits the same as the hash of the given plaintext. But by changing *n*, it is easy to increase the CPU processing cost of the sender. For the recipient's PC it is very easy to verify the result if it has the two input plaintexts and the number *n*. Let us look at an example.

Example:

HashCash [15] method works as follows: the sender computer has to find *X* in the given token: `0:date:recipient@recipientdomain:X` so that it makes the *n*-partial collision with "all zero" token. That means in practice to change *X* until hash of the whole token is having *n* leading zeros.

The token is than added to `X-hashcash:` header. Recipient can easily verify if the token `0:date:recipient@recipientdomain:X` really has a hash with *n* leading zeros. This confirms to the recipient that the sender had spent some CPU time to send the email to that particular recipient and hence it is more likely that the message is not spam.

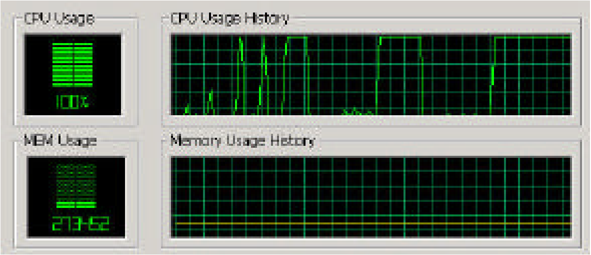


Figure 6: CPU usage when calculating partial hash collisions

If *n* increases by 1 the computation time needed is doubled. Tests on my laptop (IBM T30 ThinkPad with P4 CPU 1.8 GHz and 512 MB of memory) revealed that for e.g. 20 first bits of the hash to be zeros it needs around 3 seconds to calculate *X*. If *n* is 22, the calculation time is around 13 seconds while for *n*=25 the calculation time is around 110 seconds. Figure 6 shows the CPU usage when *n* equals to 15, 17, 19, 20, 22, 24 and 25.

This method can be fairly effective to limit the capabilities of the spammer to send a huge number of emails if the recipients would have the software that requires the hash verification.

The deployment of this approach is dependant on the recipients. Some of the leading anti-spam software is taking into account the `x-hashcash` header. It does so that the larger *n* (larger number of collision bits, in other words the more CPU time is utilized by the sender to send the message) the less probable the mail is considered spam.

This approach also has drawbacks – in large corporate email systems there needs to be a lot of dedicated hardware to do the computation. Investments in dedicated hardware are not justified as long as there are no recipients that require such a computation from the sender.

Heuristic Methods

Some email filtering products [16] are using heuristic methods to filter spam. The filtering software will examine an email upon reception and the “spamminess” will be decided after several dozens or even several hundreds of heuristic rules are applied.

For example, the software may look for missing mandatory headers in email, too many exclamation points in subject or in the body of the message, whole line in uppercase, several occurrences of “FREE” in the email body, etc.

This approach does filter a high number of spam messages. But the spammers are getting more and more sophisticated – some spammers are even tuning their messages to be able to pass several most common heuristic-based spam filters. This approach does have another problem – as spam is evolving in time, the filter administrator needs to spend some time tuning the filters so that filters do not produce a large number of false positives, yet filters remain able to filter out spam. The tuning of the filter is done in two ways – “weights” of particular heuristic rules are adjusted and new rules are added. A trivial requirement for the weight adjustment is to improve filtering. This process is not so straightforward as it requires that the administrator understands the filtering process deeply. E.g. the edjustment should not eliminate some rules unnecessaryly. Every weight adjustment should be tested usually with trial-and-error method.

4 Statistical Filtering Techniques

Statistical filters are the most sophisticated email filtering algorithms today. There are dozens of antispam software products that use some form of statistical filtering algorithms. Generally, statistical filters behave better than human designed heuristics as they take into account much more information from one email. They also have some very neat properties – to *learn*, that is, to improve filtering capabilities over time and also they are able to adapt to the personal “flavor” of one’s emails.

4.1 Bayesian Filters

Bayesian filters started to be deployed in antispam software in the year 2002-2003. It is today the most important and the most successful antispam method. Below we explain the Bayes Theorem, its importance and how it can be applied to antispam software.

Theory

Bayes’ Theorem is one of the most important theories about probabilistic analysis. Thomas Bayes was an amateur British mathematician. He was, nonetheless, the first person to publish the theory of conditional probability and the formula described below is nowadays known as Bayesian Formula for conditional probability.

If the Bayes’ Formula is used for spam filtering, we could write it as:

$$P(S|X) = \frac{P(X|S)\cdot P(S)}{P(X)}$$

In which S means “message is spam” and X is the given word or vector that represents a set of words.

So we have the formula saying "the probability that new email is spam if it contains the word X is equal to the probability that the word X appeared in a spam message times the probability of spam divided by the probability of that word appearing in a message".

When a new email arrives, every word in the new email message is examined separately. The conditional probability of the message being spam if the word occurs in email is calculated. Finally, using the Naive Bayesian formula with some number of most significant probabilities (ones that are the closest to 0 or 1) the final result is calculated.

The Naive Bayesian formula is called so because it uses the assumption that the occurrence of words in email is an independent variable. The assumption is not entirely correct, as words do not appear in the written language completely independent of each other but the assumption of independence greatly reduces the complexity of the algorithm.

How does it work in email filtering?
Here is how the Bayesian filter works in antispam software. The system already has two large tables; one contains all the words that appear in spam messages and the other all the words that appear in non-spam messages. The two tables are denoted with TSP and TNONSP, respectively. The numbers represent the number of the occurrences of the particular token in the message. Note, that it is possible that the same token appears in spam and in nonspam messages. A token could be a word, a number, or anything that is not a token separator. Token separators are usually blanks but they could be also special characters. Simply saying, a token is a *word*.
Hence there are two large tables with a long word list and in each table are the numbers of occurrences of tokens. If the system has received so far 100 spam and 100 nonspam messages, *all the tokens* from spam messages will be in the TSP table. Next to each token in the table there is a number representing how many times that particular token occurred in all spam messages.

When a new email appears, a third table is created. This table is used to process this email. For each token that appears in the new email, the filter will go through both the tables and calculate the probability that the email is spam given the token.

$$P(S|X_a) = \frac{P(X_a|S) \cdot P(S)}{P(X_a)} = \frac{P(X_a|S) \cdot P(S)}{P(X_a|S) \cdot P(S) + P(X_a|\neg S) \cdot P(\neg S)}$$

Where X_a is one token.

Note that the equation is easy to calculate from the two tables created before as follows:

$$P(X_a|S) = \frac{\text{Number of occurrences of } X_a \text{ in TSP}}{\text{Total Number of occurrences of } X_a}$$

$$P(S) = \frac{\text{Number of Spam Messages}}{\text{Total Number Of Messages}}$$

$$P(X_a|\neg S) = \frac{\text{Number of Occurences of } X_a \text{ in TNONSP}}{\text{Total Number of Occurences of } X_a}$$

$$P(\neg S) = \frac{\text{Number of Nonspam Messages}}{\text{Total Number Of Messages}}$$

The process is repeated for every token in the email.

After the table for that particular mail is constructed, the filter selects the given number of most significant tokens. The most significant tokens are the ones whose probabilities $P(X_a|S)$ are the furthest from 0.5, which means that those words are most likely to appear in spam only or in non-spam only messages.

Finally, assuming that the words are independent variables (which is not the case in a human language but we are using the Naive Bayesian formula for approximation), we can calculate the final probability as:

$$P(S|X_1, X_2, ..., X_N) = \frac{P(S|X_1) \cdot P(S|X_2) \cdot ... \cdot P(S|X_N)}{P(S|X_1) \cdot P(S|X_2) \cdot ... \cdot P(S|X_N) + (1 - P(S|X_1)) \cdot (1 - P(S|X_2)) \cdot ... \cdot (1 - P(S|X_N))}$$

Where *N* is the number of the most significant words in the email. The Email is classified as spam if the calculated probability is higher than a given threshold. The higher the threshold is, the more likely that the spam will pass the filter as non-spam. But the higher the threshold, the less false positives will be deleted.

After the email has been classified (spam or non-spam), all the words from that email are added to the TSP or TNONSP table depending on the classification.

4.2 Modifications to Basic Bayesian Filters

Several modifications to basic Bayesian filtering (described above) have been proposed to improve the reliability of the basic Naive Bayesian filtering. The reader can find more details in [17], [18].

For example SpamProbe [19] uses token pairs instead of only tokens to improve reliability. This does increase the length of TSP and TNONSP tables, however. Another approach [20] is using a sliding window of 4 words. The Window is moving word by word and in every window position all the possible combinations of words are selected using binomial polynomes. Selections are then hashed and put into a table.

Example: if the window is located in some part of the text (WORD1 WORD2 WORD3 WORD4), the selected tokens will be 1, 2, 3, 4, 1 2, 1 3, 1 4, 2 3, 2 4, 3 4, 1 2 3, 1 2 4, 2 3 4, 1 2 3 4.

All tokens are then hashed and put into the table and the window moves one word ahead. The process is then repeated until the end of the email. This improves the reliability of filtering, as phrases will be included as well as each word into the table for later spam probability checking using Bayesian technique.

HTML parsing
In order to ignore the common spammers tricks, the filter should parse the message ignoring all the bad HTML tags and also ignore all the text that is not visible in the email (white text on white background). This will make the filter ignore all the text that will not be anyway visible to the reader and is included there only to trick the filters.

Of course, not all HTML should be ignored. HTML tags contain useful information. One of the usual signs that message is spam with high probability is the appearance of `ff0000` (HTML code for bright red color). Another piece of useful information in HTML tags is `img` references. Those usually contain URLs to spammer's web sites.

One of the techniques that may be useful here that I did not find being used in today's filtering is the ratio of correct-to-garbage HTML tags. A lot of incorrect HTML tags is usually a sign that the message has been customized and bogus HTML tags are used to break word tokens apart.
Yet another useful information may be to parse the message and find the tokens that appear after the parsing and are not visible before the parsing. If the bogus HTML tags are inserted to break the tokens that have high spamminess, this could indicate higher probability that the message is spam.

Word Frequencies
Words should be treated differently if the frequencies of occurring in the email are higher. If some incriminating words are appearing in the email more then once, this should be taken into account as well. SpamProbe [19] uses user-defined numbers for *N* and for *R* (max. number of occurrences of single word).

Different Scoring depending on location of the word
Tokens are counted as different in the tables if their position in the email that has been examined is in the headers or in the body [20]. For example, one word that appears in body only may have almost neutral score but the same word appearing in the subject may have a different score.

URL, email and Subject: decoding
Sometimes spammers are using different URL encoding techniques to prevent filters from seeing them. Filters must decode all different URL encodings and return those into canonical forms. Moreover, the filters should split the URLs into parts and use every part as separate token.

One thing that currently hasn't been used is to calculate the ratio of unusual encoding in email. A high ratio of URLs encoded in different encoding techniques in the email may be a good indicator that the email's spam probability is high.

URL Checking
One interesting idea [21] speculates that the filter should fetch and check the content of the URL that has been placed in the message. The fetched page may be scored in the similar way as email content using score tables. An interesting feature of this technique will be that filters will actually increase the costs of spammers' web hosting. If each filter will go and fetch the web page content, the spammer's cost will skyrocket because of the bandwidth used. Some speculate that this approach is bordering with dDoS [22] attack, as the spammer's site will suffer from excessive traffic.

When email is not giving enough evidence whether it is spam or not, the fetched URL will give additional evidence about how the content of the web site is relevant to the recipient using the unique recipient's set of rules with the unique TSP and TNONSP tables for Bayesian filters etc.

JavaScript decoding
As spam may contain all of its content in JavaScript, the important part of filtering would be to decode the JavaScript and parse the message before running it through the filter for word scoring.

Corpus ageing
Most of today's spam messages contain a high percentage of irrelevant text either positioned in HTML tags, invisible text or at the end of the message. The effect of placing irrelevant text to the filters is destructive – filters' TSP and TNONSP tables are becoming larger and larger, and the filtering will become slower and less precise.

One approach [23] to limit this is to age out the words in the tables. The words can be aged out if they do not appear in both regular and spam mails for some period of time, or each occurrence of all the words in the table will have its own timestamp, and will age out automatically after some period of time.

The ageing time must be selected carefully and is highly dependent on the volume of incoming email messages. The tables must be accurate enough to keep most of the relevant words in emails because this is the only way to calculate aposteriori probabilities of spam. But a large number of “irrelevant” words will poison the filter’s ability to be fast and accurate. The ageing algorithm must be tested carefully[14].

Token Degeneration

Tokens in the mail can be degenerated if the token itself does not appear in email.[20] Token degeneration should improve the reliability of the basic Bayesian filtering because it can also reduce the word to its stem. (Alternatively, token degeneration may be done so that token degenerates only by non-letter token endings and letters are preserved). However, the effect of this should be examined carefully to see if token degeneration can be applied only to tokens ending with a special character or it could be applied to any token. In case of any ending, the token is reduced to its stem. (“buy” is the stem of “buying”, “buyer”, etc…)

5 Hybrid Methods

5.1 (Bayes+whitelists+RBLs+other)

Hybrid methods for filtering can be applied to improve the accuracy of the spam filtering. For example, whitelists are applied to prevent filtering if the sender’s email address is known to the filter. Some commercial software is using this technique [25], [9], and is also automatically maintaining whitelists to reduce the end-user work. RBLs are not so accurate if used as a sole technique. But if the lists are combined with probabilistic filtering, the result may be better. An email is more likely to be spam if it comes from a domain listed on RBL but it does not have to be spam automatically. This could be taken into account in the final decision after the statistic filtering is applied.

Also, heuristics combined with RBLs and probabilistic filtering may further improve the decision process. Heuristics themselves can be given “weights” based on their accuracy and relevance. This can be implemented using a separate table that will change the heuristic weights based on aposteriori filtering results. The fine-tuning of the filters and finding the right balance may be a very hard task.

5.2 Other Anti-Spam Methods

Other anti-spam methods that do not belong to the categories described above will be briefly discussed next. Typically, these methods complement anti-spam methods.

There are lots of other methods for dealing with spam. One of the most interesting ones is “Slashdot action”, in which the community reacts to punish the spammer by spamming him with junk-mail.

More sophisticated approaches use spam filtering based on neural networks, text classification, other statistical algorithms, etc. Although some approaches may be very useful in dealing with the spam problem, their usefulness is yet to be confirmed.

Education

One of the recent Yahoo! Surveys [14] shows that 48% of the people actually believes that by carefully opting out when spam is received will help them reduce the number of spam messages they receive. This shows that most of the email users do not really understand the techniques that spammers are using. Educating the users to avoid opening, reading or responding to spam email is the most important measurement to prevent this “Internet cancer” from spreading. Considering the fact that this year is perhaps the first year when general public’ interest is turning to the spam problem, it is of greatest importance that general population is educated about it as soon as possible.

6 Side-Effects

The unwanted side effects of anti-spam software are discussed in this Section.

6.1 False Positives

False positives are very problematic. In essence, those are email messages that shouldn’t have been filtered but they have. Although the best products can deliver a number of false positives well below 1%, still it is too much for some users and especially for corporations. The very possibility of this kind of errors is even prohibitive in some cases. Therefore, prior to any anti-spam software deployment the pros and cons should be understood very well.

[14] Editor’s Note: It seems that irrelevant text confuses the probabilities of words rather than the words themselves. It follows that ageing should be used to occurrence values, not to the words themselves.

6.2 Freedom of Speech and Government Monitoring / Government Enforcement

It is important not to forget that the concept of the Internet in general, and email in particular, was designed, is used by and meant to be free for anybody. It has been used as a cheap, fast and reliable means of communication. It has improved greatly the business correspondence as well as private communications. Spam is just one annoying side effect of having such a great tool.

If there will be a legal requirement for ISPs to implement a spam filtering tool, then it will become up to ISPs to *open* and *examine* one's email and to decide whether the message is appropriate for the recipient. This is a questionable right for the ISP. But lots of today's ISPs are doing so.

If, sometime in the future, a government body is able to update the ISPs filters in order to prevent spam, it is not hard to envision a dark scenario in which the government of some country updates ISPs filtering rules in order to prevent email correspondence that is not in line with the current political standpoint of that particular government.

It is of *crucial importance* to allow the free circulation of email, so that *each email* is able to reach *each destination*. And it is up to the end-user to decide what he/she is interested in and what is spam. If anyone else is deciding for you what is the appropriate content that you can see, this may be misused in ways that are even hard to predict.

7 Conclusion

The spam is growing at a tremendous rate. If nothing is done, email will be useless in a few years. Luckily, anti-spam state-of-the-art is gaining momentum and some very promising tools are appearing on the market. Currently, the most used anti-spam techniques involves Bayesian statistical filtering together with heuristics and optimizations. Corporations are also looking for ways to prevent spam reaching their employee's inboxes and are more than ever interested in efficient anti-spam methods.

In order to really stop spam, there are several steps to be taken:

1. Educate, educate, educate. Everyone should be educated on what is spam and how one can prevent it. People should be educated how to download and use anti-spam software. More importantly, the general population should be educated not to answer nor to reply to spammers and never to buy from them. Improving the general level of education will increase the usage of anti-spam tools and will subsequently raise the spammers' cost.
2. Major email client vendors should deliver their products with at least statistical filtering options by default. And the most important – it should be up to every user to train his software to recognize spam because everyone's rules are different.
3. ISPs should not be the ones to filter the emails from spam because this reduces the control of the end-users. However, ISPs should enforce email use policies and install software that prevents spam originating from their domains. Alternatively, ISPs could filter emails based on subscriber's preferences if authorized to do so by the subscriber.
4. Laws against spam should be very carefully examined, and should not allow governments to act proactively but only to reactively prosecute spammers. Laws should be written not to allow sending mass mailings to users that did not have previous business relationships with the sender.

References

[1] "Spam in the Enterprise: Market Problems, Needs and Trends", May 9, 2003. http://www.ostermanresearch.com/whatwedo.htm

[2] Mail Abuse Prevention System

[3] Cobb, Spam By The Numbers, (Source ePrivacy) http://cobb.com/spam/numbers.html

[4] ePrivacy Group: "Spam: By The Numbers", ePrivacy Group 2003. www.eprivacy.com

[5] Junnarkar, Sandeep: "AOL touts spam-fighting Prowess", 30. April, 2003. CNET News.com, http://zdnet.c om.com/2100-1105-998944.html

[6] Multipurpose Internet Mail Extensions, Draft Standard, www.ietf.org

[7] Graham, Paul: Stopping Spam, August 2003, http://www.paulgraham.com/stopspam.html

[8] Overly, Michael R: "Email, Adult Content, and Employment Law: Reducing Corporate Liability With Filtering and Policy Tools, Foley & Lardner, 2003.

[9] Nokia Message Protector, www.nokia.com

[10] www.postini.com

[11] Leung, Andrew: "Spam – The Current State", August 2003, Telus International

[12] "E-mail glitch blocks Harvard acceptance e-mails", February 2002, http://www.usatoday.com/tech/news/2002/01/02/harvard-spam.htm

[13] Gaudin, Sharon and Gaspar, Suzanne:, "The Spam police", *Network World, !0 September, 2001,* http://www.nwfusion.com/research/2001/0910feat.html

[14] Yahoo! Anti-Spam Fun Facts, http://antispam.yahoo.com/funfacts

[15] www.hashcash.org

[16] Spam Assassin, http://spamassassin.org

[17] Graham, Paul: “A Plan for Spam”, August 2002. www.paulgraham.com/spam.html
[18] CRM114, http://crm114.sourceforge.net/
[19] SPAMPROBE - http://spamprobe.sourceforge.net/
[20] Graham, Paul: “Better Bayesian filtering”, January 2003., www.paulgraham.com
[21] Graham, Paul: “Filters that Fight Back”, August 2003. http://www.paulgraham.com/ffb.html
[22] Graham, Robert: “Hacking Dictionary”, http://www.robertgraham.com/pubs/hacking-dict.html
[23] POPFile – www.popfile.sourceforge.org

Economical Impact of SPAM

Timo Ali-Vehmas
Timo.Ali-Vehmas@pp.inet.fi

Abstract

Unwanted Email is a phenomenon created by the fast growing Internet business bubble of late 1990's. It however did not disappear with the vaporising bubble but it stayed with us and has grown faster than any other service in Internet ever since. Unwanted email is now reaching a mature state representing up to 50% of the overall email traffic. Unwanted email is a complex technical and primarily economical issue, which may, in the worst case, lead to lower use of electronic services in general. Unwanted email, called also spam, has a possibility to become a Real Killer Application to the Internet, not in the Internet.
The economic impact of the spam is difficult to estimate quantitatively. Instead, a value system based analysis is carried out studying the real and potential value generation in the networks of spammers and their partners. This is supplemented with selected pieces of quantitative information about the volume of the business, merely as examples. There are several studies in the literature how spam is impacting various value chains but there are very few references found for the spam value chain or system itself.

Key words: unsolicited email, spam, ecosystem, value chain.

1 Introduction

Internet is today the most versatile communication network in the world. It serves it users well in a large variety of applications, ranging from banking to gaming and from browsing to emailing. There were some 300 million users in the Internet in 2001 and the estimate for today is over 400 million ranging up to 125 million in the USA alone [1]. The Internet penetration level is actually highest in Europe, Nordic counties and the Netherlands leading with about 60 % penetration but there is no direct correlation the to penetration of spamming. Email has become one of the most important applications primarily because of its quite good interoperability and compatibility between different service platforms. Simple IETF specifications for email, such as Simple Message Transfer Protocol SMTP (IETF RFC 788), Post Office Protocol POP3 (IETF RFC 1081) and Internet Message Access Protocol IMAP4 (IETF RFC 2060) [2] and their extensions have been developed in an idealistic research environment where malicious use of Internet has been almost a capital crime as a starting point. This approach has left email without proper protection against users who may have a different starting point and ethics than the research community. Another factor promoting wide use of Internet in good and in bad is the billing mechanisms, which do not separate uplink and downlink traffic and where all subscribers pay for both incoming and outgoing traffic. Further on with broadband access the tariffs are mostly flat or block rate based. This leaves the door open for anybody with very low entry fee to enjoy all the great benefits of the Internet including email with no feedback measures irrespective of whether the use is economically justified or not.

2 What is SPAM ?

Spam originally meant “Spiced Pork and Ham”, a canned pork meat, which was not allowed to be marketed as real ham because of too low high value content, i.e. ham. Internet community adopted the term from Monthy Python Flying Circus where spam was part of every meal of a restaurant whether the customer wanted it or not. This is a very good simplification also for a much more serious business issue of today's Internet, the Unwanted Email.

Unwanted Email is not simply all emails that people receive unsolicited but it may be categorised better by dividing it up to three groups:

2.1 UCE and UE = UBE

Unsolicited Commercial Email (UCE) means emails that have been sent to the receiver in order to advertise products or services. The actual sender of this email may or may not be the same body as the retailer of the advertised items. But not all the unsolicited commercial email is spam. It may well be that the receiver has earlier permitted his or her email to be addressed by commercial advertisements. According to the current directives in EU 95/46/EU 97/7/EU and 97/66/EU such email advertisement is legal.

Directive 2000/189/EU goes further defining for email and also for the GSM Short Message Service that only Opt-in scheme may be used. Similar legislation is either available or being prepared in other major markets, Japan and the USA. Currently in the UK UCE is not allowed to consumers but is still allowed to corporations. [26]

Unsolicited Email (UE) may be spam even if it is not commercial. Also political and religious advertisement is regarded as spam. PEW Internet & American life project has recently published a large survey about spam [3]. According to the survey, people are quite sensitive to spam today. As high fraction as 74 % (with the error margin of 4%) of people consider even a personal or professional email from a person they do not know to be spam. Unsolicited Commercial or other Bulk email is also referred as UBE.

A clear difference is visible in this study to show that only 11 % of the interviewed people considered unsolicited commercial email as spam, if they only had given the permission for such transmission in advance.

Hence, there are two concepts of sending Unsolicited Bulk Email, which shall be recognised clearly separately.

- Opt-In. There was a permission given in advance by the receiver to the sender to send commercial or other emails, automatically. This should not be considered as spam

- Opt-Out. There was no permission given by the receiver but there is a reliable mechanism for the receiver to forbid such transmission for the future. This is not to be considered spam, necessarily.

Some member states in EU, including Finland, have implemented the Opt-In scheme in national legislation already several years ago. [4]

2.2 SPAM

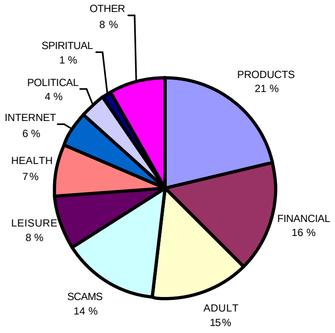
But the problem really is when the Opt-out request is not used or not taken into account. This is the case when we really are talking about spam.

3 Market of SPAM

Different businesses utilise spam differently. Proportions of spam advertisement in different businesses and markets give some indication about the losses because of spam. The total value of spam-based business is difficult to estimate.

There are many different estimates of what is the content of spam but with a rather large error margin they all agree. The top 3 categories are always product business, financing and banking and the 3rd one, adult entertainment. Some estimates show that the share of SCAM, i.e. Nigerian chain letter -type swindle is also quite remarkable, which in other estimates may be included in te financial category.

The estimate of Figure 1 is provided by Brightmail, an anti-spam company that is one of the most active participants in the global debate about spam and its consequences. The Anti-spam companies are discussed in detail in Section 5.7. [5].



One view to SPAM Content, Source: Brightmail, 11/2003

Figure 1. Content of spam, Source: Brightmail.

Another way to look at the market of spam is to study in what countries spam is most wide spread (See Figure 2). Currently the USA is most vulnerable to spam by far. The USA represents probably one 3rd of the total Internet users but for spam, its market share is almost two out of three. It would be a good study item to research what are the factors in the USA that are making it so vulnerable to spam. [6].

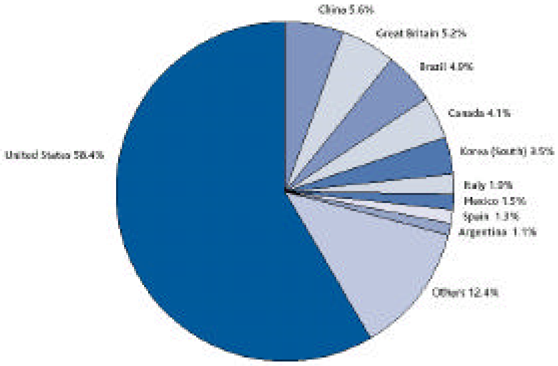


Figure 2. Markets of spam. Source: MessageLabs

4 Volume of SPAM

There is a lot of information available about the growth rate of spam in the recent years and months. There are also several estimations what this all means to the users and the Internet Service Providers.

4.1 How did we get here?

Email as a broadly recognised phenomenon started early 1990's. By the middle of the decade it had been adopted by all major and also many smaller enterprises, Universities - where it all began, and public authorities. The general public was not yet exposed to email over the Internet until the great IT industry stock market bubble started to emerge. Still in 2001 according to an estimate by Brightmail spam was only 8% of the total email traffic but already in 2002 it reached 30% of the total email traffic and for 2003 it is claimed that spam emails exceed the number of ordinary emails in the Internet [5]. These figures must be viewed with some criticism. Most of the estimates, that were available for this study, were provided by the firms developing tools and services to reduce spam i.e. the so called anti-spam companies. Some estimates [7] can be interpreted even so that the normal email has gone down because the total absolute growth rate is lower than the absolute growth rate of spam.

An independent market research company, IDC estimates that in 2002 the proportion of spam was 18% of the total email traffic, which still is a considerable 5.6 Billion spam emails every day. Also IDC estimations of the growth rate of spam are more modest than e.g. Brightmail's estimates showing some 20% growth for spam and 15% for normal email. This would keep the spam figures for 2003 still below 20% of the total email traffic. [8]

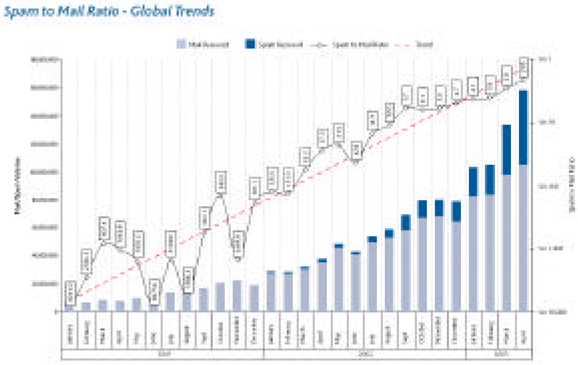


Figure 3. The growth of email and Spam. Source MessageLabs

The estimates of a UK based anti-spam company, MessageLabs, are somewhere in between (See Figure 3). These estimates show the growth of email and the growth of spam in a very comprehensive way. [6]

There is also one additional element here, which may impact these estimations. In corporations and other large communities, a major part of the email is internal, within their own domain. This email is only occasionally, in case of a virus attack, polluted by spam. Therefore, corporate email users typically see spam only as a percentage of incoming "external" emails, not as a percentage of all received emails. This is clearly having a major impact on some estimates.

Hence, one origin of differences in estimates is whether the question is of all email or email from the Internet.

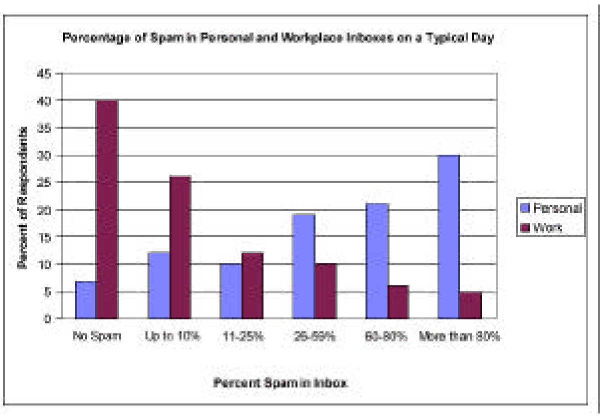


Figure 4. Consumers and Corporate user experience of spam. Source PEW Internet & American Life project.

To summarize our charcterization of unwanted email as a phenomenon today, it is easy to agree that first of all, it is a severe problem, its growth rate is significant and it is maturing as a business.

We need to separate real spam from controlled unsolicited email. This is today in practise an impossible task. Therefore, this study tries to address both and to indicate also some common elements of these two.

5 Value system

Value chain normally means the overall flow of material or immaterial added value, where value flows downstream and money flows upstream. In many businesses a flow is a far too simple model since there are many indirect links and sometimes money flows also downstream in the form of subsidies. Therefore, also in this analysis a different term, value system is used, instead. Also Ecosystem is a term used for a similar purpose.

The real issue, however, is that the overall value system related to spam is very fragmented, not too well understood and also partially underground. There are several commonalities with other clearly illegal activities such as money laundering. Therefore it is very difficult to get accurate quantitative figures that would be comparable and that would provide a comprehensive base for analysis. That is why this study is focusing on the value system itself. This approach may add more value than analysing some part of the value system in great detail. It is important to understand the overall value system and the interrelations between the players in order to even estimate the economic impact of spam. Very few previous studies cover the overall value system of spam [25].

One hopefully usable side effect of understanding the value system may be finding some means to fight the real spam, as a separate item from unsolicited emailing, which still in many cases is not spam.

The value system of unwanted email includes some fundamental players. They are:

- Spam hosting including
  - Address generators
  - Content generators
  - Full service providers
- Spammers and their supporters
  - Spamming Software vendors
  - Hackers and hacked computers
- Legal UCE advertisers
- Various ISPs on the sending side
- Network operators
- ISPs supporting receivers
- Corporations
- Consumers
- Product and Service retailers who finance the UCE and also spam.

In the following Sections we will discuss the role and motivation of each one of the players. We can show that most of the players are players against their own will. We may call them victims but surely some of the players have a very strong role in driving the use of email in advertisement and unfortunately, some of them do it ruthlessly, abusing the resources of others.

5.1 SPAM hosting

The Spam hosting community is a very interesting part of the value system, which is at least partially underground, like the roots of a tree. Spam hosting includes a large network of different kinds of Internet oriented small firms and individuals who earn their living by providing primarily content and address data bases for the actual advertisers, spammers or others.

5.1.1 Address database aggregators

Address database aggregation and reselling encompasses a complex network of players who create and develop the email address databases based on various mechanisms. The most visible mechanisms include:

- Web portal clicking and related enquiry of email addresses and other contact information,
- Search engines to look for Homepages and Newsgroups and email addresses on those,
- Aggressive bulk email harvesting attacks (considering all random email addresses that do not pounce back to be real),
- Aggregation of the address databases created by mechanisms mentioned above and combining these with e.g. Opt-in databases of their customers,
- Segmentation of the databases on geographical, ethnic, habitual etc. basis,
- Reselling the databases, providing subscriptions to a continuous database service.

Addresses are available at a very low price, between $3 and $100 (or Euro; the estimates are very rough) for one million email addresses [4]. Taking into account that there are only some 600 million email addresses, the total value of the whole Internet email address database would be between 1800 – 60000 USD. Without any added value, such as a very good segmentation, at the end of the day the business opportunity of bulk address processing is very small. It is likely that the content of these address databases is in most cases very poor and actually do not generate spam from the receivers' perspective because the emails reach nobody. This type of spam flooding still loads the transport network and the receiving email servers badly. When taking into account that the number of major spammers is only a couple of hundreds by some estimates, the potential customer base for simple address aggregators is also quite limited.

5.1.2 Content creation

Content creation and aggregation for spamming is another partially underground activity. In legal unsolicited advertisement the content is directly generated together with the retailers. There is, however, some evidence that some retailers are using the spam content creation and the spam hosting network as a decoy. The Spam hosting network or firms generate faulty content that is based on false claims and false information in general which is then placed on several web portals as baits. When consumers respond to the bate, the network collect the email and other relevant information. But they never deliver anything since there was probably nothing to deliver in the first place. But somehow, through several steps like in money laundering the address information finds its way to the legal retailer or service provider who can use this well qualified contact information for a related business offering [9]. A report by the US Federal Trade Commission (See Figure 5) claims that 66 % of all spam has some false information in either sender address, subject field or in the text part. The value varies between 44 % contain false information in product oriented advertisement up to 96% false information in advertisements offering investments and business opportunities. [10].

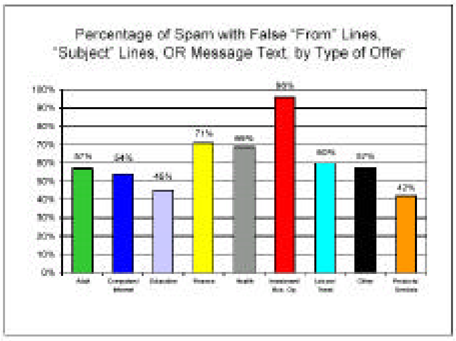


Figure 5. Portion of spam containing false information, Source US Federal Trade Commission.

This points to a strong invisible network in contact information harvesting but it is very difficult to estimate the total volume and financial importance of this partially underground business. It may well be that the address database market is only a tip of the iceberg where not so valuable email addresses are sold to email spammers at very low price but the really valuable contact information with additional contact and profile information on the owner of the email address is used for more sophisticated direct marketing, ranging from banking to time sharing free time and vacation offers.

5.2 SPAMmers

Spammers typically shall be discussed separately from spam hosting. Spammers are the actual organizations or individuals who push the button and send the spam email flooding to the network. There are several different types of spammers, some are well known individuals who have several Internet accounts and who use those accounts directly and openly to send the spam. They may be occasionally blacklisted in one IP address or another but they soon pop up from some other IP addresses. Detroit Free Press 12/2002 claimed that "spam king" Alan Ralsky operates 190 email servers to send his messages.

Some of the spammers may at least pretend to use Opt-out registers and some may even use them. There is one estimate claiming that the majority of the openly but well organised and operated spamming may be driven by no more than 200 different parties or persons. [9], [18]

As an example of another kind of spammer we could look at *"Ms. Betterly who quickly discovered that she could make a profit if she got as few as 100 responses for every 10 million messages sent to clients, and she figures her income will be $200,000 this year".* Ms. Betterly was interviewed by Wall Street Journal in November 2002. [11]

The fatal type of spammer who probably is the most difficult one to take under any control is the one who actually use viruses and other hacking methods to hijack unprotected computers. These people spread their spam email quite often without any commercial or other purpose. Their only aim may simply be to cause maximum harm to the selected receiver or receivers or to the overall Internet. A major part of the commercially oriented spamming takes place with this approach too. Some estimates are claiming that up to 70% of all spamming goes via hijacked computers [9].

5.2.1 SPAM Software vendors

A dedicated group of software developers is giving a helping hand to the spammers, many of which are just ordinary opportunistic people. These software developers have talents in email software but also in Internet technologies at large. Some of the earlier hackers are using their experience in this less risky way. There is no statistics available on these vendors.

5.3 Legal direct email advertisers

At this point we have to discuss also about another group of bulk email senders. In 2001 there were about 50 companies openly offering services for electrical direct marketing. These are sophisticated enterprises that typically have a full service approach with people and tools to serve their customers. But what is very important, these companies typically use the rather unreliable Opt-out approach to limit the really annoying amount of email. Also the address databases are supposed to be of high quality. [4]

One company, 24/7REALMEDIA, advertises on their Web page: "Our products and services include our patented ad serving technology, **Open AdStream**®; web analytics via

Insight XE^{TM}; full service search engine marketing programs via **24/7 Search**; integrated online media packages as well as Web site representation via the **24/7 Web Alliance**; **online promotions**, **email marketing**, and direct to desktop solutions via the **24/7 Messenger**."

The 24/7REALMEDIA and its kinds have recognised by now that a successful business relation requires trust and trust can be built only with reasonable business ethics. Using Opt-out and/or Opt-in approaches the direct marketers achieve actually better sales than with massive wasteful spam campaigns. [12].

It is important not to mix these companies with spamming. As seen in Section 2, very few email receivers, 11 % consider Opt-In email direct marketing as spam. There are eMarketing training and consulting firms available, too, which fortunately, at least in public messages, strongly encourage their clients to use Opt-In approach.

"Opt-in mail is more personal. You can personalize your message to each recipient. Third, opt-in means that the recipients have chosen to accept and read your messages. They're interested in the information you are offering."

What is even more important is that most of the side effects to corporations, network operators and ISPs, as we will discuss in Sections 5.4 to 5.6 are totally avoided.

When estimating the impact of spam the direct email advertisers should not be included into the calculations.

5.4 Internet service providers

Internet service providers are the key group in the spam value system in many ways. First of all there are ISPs such as TeliaSonera that recently suffered concrete damage because of virus based spam attacks. The direct costs involved were only about 3 M€ but it is very difficult to estimate the value of all the bad will and publicity TeliaSonera received and what finally was the opportunity cost of lost old and new customers.

Economical impacts to IPSs include wasted memory and server capacity, wasted network capacity and nowadays more and more, capex and opex of special servers to filter and mark the incoming emails for the protection of their network and customers. Major ISP's such as Time Warner (AOL) and MSN claim that they filter and block 2.4 Billion emails per day each. This may represent up to 80 % of all incoming traffic.[13]

It is obvious that free web email accounts really are the worst to receive spam because no commitment is required from the mailbox owner to open such service. Naturally these mailboxes may also be used to gain an anonymous identity in order to subscribe to some further questionable services. All of this behaviour is increasing the likelihood of receiving spam.

Total economical impact to IPS's is difficult to estimate but one claim by BellSouth is that there is some $3 - $5 cost penalty per each Internet subscriber.[14]. Assuming 400 million Internet users [1], this would top up to $2 Billion. It may be more reasonable to scale this down to cover mainly USA and maybe take the lower end of the estimate, too. Still the wasted effort is as high as $400 Million per month or about $5 Billion per year.

There is the dark side of the coin too. It is quite likely that some ISPs are in a deeper business relationship with the spammers. There is some evidence that some spammers have paid quite high fees to their ISP's. These "pink contracts" are kept well confidential and therefore the actual amount of money is very hard to estimate. But in most of the cases this can only be a fraction of the spammers' overall revenues and therefore for now let us consider it as just a minor interesting detail. This however is one important element when analysing the overall value system. It is more and more obvious that this quite a small business, which spamming itself is after all, causes considerable harm to innocent Internet users and service providers. [9]

Naturally, in case of legal direct email marketers it is obvious that there is a value and money transfer between them and their Internet services providers, but again, this is not part of the economic impact of spam.

Unsolicited Email is a real problem in wireless industry only in Japan where the leading wireless network and service provider, NTT DoCoMo, has suffered from I-Mode spam for several years. According to some estimates, the damage to DoCoMo is of the order of $200 million. This is a significant amount of loss but is still relatively small when compared to the overall losses caused by spam for the wireline service and network operators. [26]

5.5 Network operators

Network operators are a group of players who simply pass the traffic through their backbone networks. Again it is very difficult to estimate the economical impact but taking into account the low real time requirements of email traffic and operators capability to differentiate real time traffic and best effort traffic at least in the ATM backbone, we may assume that this kind of data transmission is still only a modest share of the total best effort traffic and is not able to severely threaten the backbone network operators. In many cases network operators get also positive revenue based on the traffic the ISP and corporations generate and therefore we may assume that the economical impact may actually even balance out for network operators.

In some special situations, e.g. a massive virus based, scheduled email attack may cause overloading also in the backbone network and Internet root name servers. In some cases, the damage caused by one individual email worm may have exceeded 1 BUSD. This is quite a significant amount of money but it may not be appropriate to include this to the overall calculations due to the attackers' quite different, almost terrorist behaviour. Also the purpose of these types of email attacks is more to simply cause damage to the Internet itself rather than try to gain any form of profit.

5.6 Corporations

An independent market research firm, Radicati Group, estimates in its study, Anti Spam Market trends, 2003 – 2007 that corporations worldwide have to spend up to $20,5 Billion in 2003 in servers and related operations in order to fight the incoming spam. It is unclear how much of the lost productivity is included. This may grow over $100 Billion by 2007. A separate study by Ferris Research proposes that lost productivity because of spam emails in USA in 2002 would be $8.9 Billion. [13]. It is unclear if the two figures overlap or are complementary since if the tools the corporations are using are effective, the lost productivity should be minimised. Some studies suggest that spam filters reduce the number of employees who suffer from spam from 19% to 5%. This would indicate that some portion of the wasted effort should still be included.

Anyway, this is clearly the highest figure of economical impacts listed in all material available for this study and hence it can be argued that the corporations are by far the biggest loosers due to the global flood of spam.

It is also important to note that anti-spam equipment and services may be quite costly. This leaves a large number of small enterprises to a really difficult situation. They must carefully decide what is the least costly approach to deal with spam, let it come through or acquire some anti-spam equipment or software, disconnect from Internet totally - or simply go out of business. For the consumer it is possible to abandon an email address when it gets badly infected but an enterprise email address is typically connected to brand value and changing the email address is not so simple.

5.7 Anti SPAM companies

A small portion of this great spending by the corporations and also ISPs goes to the emerging hot business of anti-spam companies that provide sophisticated tools and equipment to fight against spam.

This industry did not even exist a few years ago but today its total revenue is estimated to be $650 Million [13]. Radicati Group has predicted that it has the potential to grow to over $2 Billion by 2007 if spamming is not limited or reduced by any other means. There are now some 20 to 30 companies providing services in this business domain. Brightmail, who claims to have 11% market share and also that it is protecting some 300 million customers of ISPs is one of the most visible one. Like the spammers, also this group of companies has its roots in the big Internet bubble and it seems to have stuck with the bubble time public message. It is hard to believe that all their claims are fully reliable. But again, it is more interesting to look at the behaviour of the companies and their role and connections in the value system of spam rather than to be precisely right with the figures about them or the figures they let out.

5.7.1 Technology Insight to anti-SPAMming

Spamming is based on the very basic technologies of Internet as such. No novel technology is needed for spamming. However, the companies fighting against spam have developed several new approaches to this problem. It may be interesting to look at some of these even if it is not absolutely mandatory for an economically oriented study like this.

Several different technologies are applied to build the servers, databases and management processes of the anti-spam companies [15]. The simplist methods use just black or block or white listings of the sites known to spread spam. This however is not very efficient and causes many problems because of false denial of service incidents. Also, in the beginning simple finger prints or signatures were used as evidence of spam, which lead to many false alarms. Using some collaborative listings the fingerprints and various listings can be developed further. But all in all these technologies are used today only selectively as a second priority. [18]

Bayesian string filter

The novelty is in the way the spam mails are detected from the normal stream of emails. So called Bayesian filter string classification is used today in most of the filters as the core technology. The filter is adaptive to both spam and non-spam emails and their characteristics. The filter is also customer specific. This is important because each victim of spam has different categorization what is spam and what is not. This is also where the biggest advantage is also over simple site black listing.

Best Bayesian string filters can converge quite well with only hundreds of emails. Training may be manual or training can be done in advance based on a larger set of emails. The final novelty is that these filters will tune themselves to filter customer specific spam avoiding the problem of one filter does not fit all. Bayesian filters for spam protection were first introduced by Microsoft research and by Pantel and Lin in 1998 in the AAAI-98 workshop [17].

With later enhancements it is possible to achieve six nines accuracy, typically with zero false positive detection, i.e. one error per 1 million emails screened.

Squelch Spam email on protocol level

Instead of a simple black listing and blocking all the emails from a certain source address, it is also possible to delay the email protocol. This would cause a lot of reduced performance to the sender of spam [17]. This technology adds some costs per message also to the sender of spam while keeping the legitimate email untouched. All the emails, including those, detected as spam can be finally put through to push the false positive detection to zero.

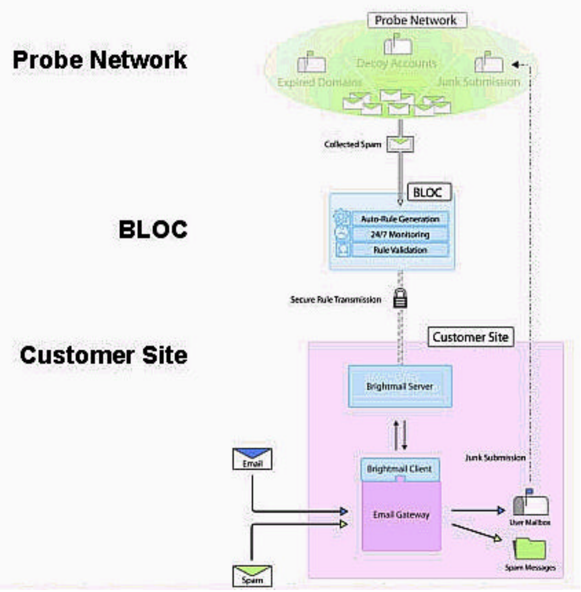


Figure 6. Brightmail patented spam filtering system. Source Brightmail [5]

Brightmail is using a special probing network, a fairly large set of email addresses opened up for this purpose only. They get a large incoming flow of emails that in this case, all should be simply spam. In their back-office they calculate detection patterns based on the characteristics of emails. They know that all the information in a typical spam message is unreliable as such, it will vary from sample to sample, even within one flooding but they use this method to collect input training data for the actual spam filter servers connected to their customers' email servers.

There must be a real time connection between the customer email server, Brightmail server and the Brightmail back-office because one spam flooding typically lasts for quite a short time. The time from the first detection of a new spam mail and when the first similar email arrives to their customer system is always very short and if the probe network is not a competitive decoy, it may well be useless. In this business time really is money.

Brightmail is using traditional customer feedback as an additional tool to pick up the spam mails that were not caught by the probe network. It is also obvious that there may be some "not-so-spam" emails that each customer may want to include into the filter traning data.

Haiku

End users may also add some specific detection part to all their emails, which will cause strong positive non-spam convergence in the spam detection filters, regardless if those are traditional or more sophisticated. Whether this really provides a long term solution is maybe less important. But it will help cultural and ethic diversity to spread. Most of these specific pieces of text are poems or proverbs or similar.

5.8 Consumers

The Consumers are the big question mark in the value system. Several studies clearly state the consumers are very much against spamming as discussed in Section 3.

Still the same studies show that up to 7 % of the interviewees have in fact ordered a product or service that was advertised in a spam email [3]. Further on, the same study proposes that in USA in 2003 some 44 % of all the email accounts are without any spam filter.

When adding the ignorance on how to protect the personal email address in order not to get on the lists of spammers it is obvious that the market is easily created. When only

0.001% positive feedback is enough to keep the flood of spam emails pouring in, the equation is ready: one hit per user per 7000 spams in the inbox, one could claim. The vast majority of the consumers are suffering because of the ignorant or reckless behaviour of others.

We should also remember that consumers' mailboxes look quite polluted because they most likely receive far less real email than the corporate users. Therefore, the percentage of spam in consumers' inboxes looks much more severe than it actually is as we discussed in Section 5.

How much extra this will then cost to consumers? It definitely depends on the connection type the consumers have. In Section 5.4 we estimated the added costs to the ISPs, which naturally have to be paid by their customers, most of them being consumers. Additional cost may incur if time or volume based charging is used for the consumers access connections in case of PSTN, ISDN or wireless. This cost could in theory become quite significant but I assume the consumer to change his email account should it get too much loaded. Therefore, I tend to believe that consumers' costs are of the order of the ISP's expenditure for anti-spam servers and additional hardware and software in general.

If we compare the success rate required by the ordinary opt-out or No-opt spammers with their cheap address lists and the good screening level of the modern Bayesian filters, this gives some hope that the commercial spammers may not any more be able to reach their 10 per million success rate. This would have an impact on a major part of the spamming value system but would still leave the door open to the plain attackers whose only motivation may be to cause harm to the Internet and its users.

5.9 Retailers using UCE

At the end of the value system are the great profit-mongers of spam, who use it for their marketing campaigns and for many other purposes.

It is important at least to try to estimate the business volume based on spam emails. This is very difficult. There are however, some estimates available for the business of adult content based on spam and also for SCAM, which are of the order of $2 and $3.2 Billion respectively. If these two represent some 15% of total spam each, one could estimate that the overall value the consumers are spending should be of the order of $15 Billion. This estimation however is very unreliable. Especially banking and financing sector estimates would have been quite interesting but this part of the value system is also least visible. [13]

5.10 Value system of SPAM

Finally, the overall picture of value system of spam can be shown, see Figure 7. First of all it shall be noted that this graph includes both legal and less ethical players in the spam related value system. By far not all the connections between all the players are clear. The picture that we are able to put together does not imply that all the players that have some red colour operate un-ethically, it rather implies that among these players there may be some who do.

At the end of the day the picture can be interpreted also as a tree, with its roots underground, trunk transporting the value to the leaves and flowers and then finally the fruits are eaten by the harvester. More detailed analogy however is not applicable.

The role of anti-spam companies is anyway interesting because they may be able to fight the spam better than expected but at the same time they will spoil their future growth potential. Spamming in the future may more clearly be divided to pure Opt-in direct email marketing and then on the other hand to plain Internet terrorists who simply send garbage email in order to spread viruses and cause harm.

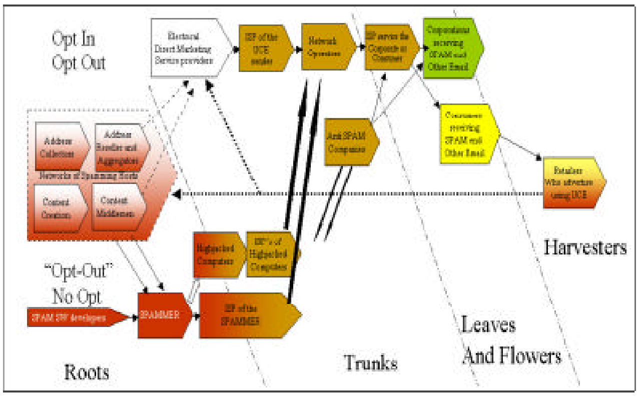

Opt In
Opt Out
Electronic Direct Marketing Service providers
ISP of the UCE sender
Network Operators
ISP serving the Corporate or Consumer
Corporations receiving SPAM and Other Email
Address Collection
Address Reseller and Aggregation
Networks of Spamming Hosts
Content Creation
Content Middlemen
Anti SPAM Companies
Consumers receiving SPAM and Other Email
Retailers Who advertise using UCE
Highjacked Computers
ISP's of Highjacked Computers
Harvesters
"Opt-Out"
No Opt
SPAM SW developers
SPAMMER
ISP of the SPAMMER
Leaves
And Flowers
Trunks
Roots

Figure 7. Value system of SPAM

Nowadays there are naturally many other actors in the spam related matters. These include legal people, lawyers and authorities, news agencies, market and other research organizations and so on. But the overall economical impact of spam is still considered moderate to these businesses. Therefore, we have omitted detailed analysis of these actors in this study.

6 Economic Impact of SPAM

Based on the discussion above the overall money circulating in the value system can be as high as $40 billion, including the expenditure of corporations, business value of spam related sales of goods and services and some additional costs for consumers, ISP's network operators and others. The summary is shown in Figure 8.

In order to put this to some reference, at the same time the overall retail business in the USA is about $3000 billion. If both figures are even roughly right, the economic impact of spam is significant.

It should be noted also that the losses for the corporations most likely exceed the value that is generated by the parties utilising spam. It actually would be cheaper for corporations to buy all the adult content, respond to all Nigerian chain letters and get some "very advanced weight contol gadgets" for their employees. What a waste!

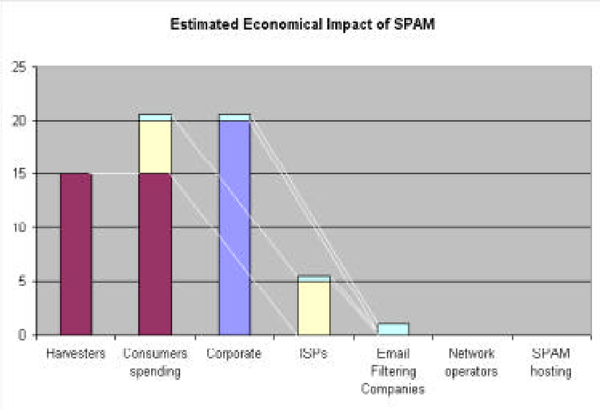


Figure 8. Estimated Economic Impact of spam

This unbalanced equation is there because the cost of sending spam is so low and most of the costs incur to the receiver. In order to be able to fight successfully against spam, this equation has to be changed.

Recently, November 22, 2003 similar results were published by Untad, [27] noting also the large variations of the estimations. It is important to take the absolute figures as rough estimates only because of so much of the value creation and loss is not reported and takes place "under ground".

7 How to avoid SPAM in the future?

It is obvious that technology solutions alone will not be able to stop all the spam flooding to our mailboxes. Changing the equation discussed above would have a major impact to the majority of the spam. Spam filters based on the black and white lists do not work but supported by novel test classification technologies such as Bayesian filters will improve the quality of filters to the level which makes the low quality bulk spamming uneconomical. Also, if we can misuse, even temporarily some email protocols to put some extra burden also to the senders of spam this would help turning the equation right.

In order to limit the spamming based on hacking methods, one reasonable approach is to make email virus scanning first recommended but later mandatory for the ISP's. All emails containing some virus or worm should be stopped already before they reach their target computer. Consumers are not very well aware of all the risks in the Internet, they should be protected reasonably well by the service providers. Since consumers do not use spam filters nor virus protection, these obviously should be tasks for service providers.

In long term it is also possible to develop better email protocols to include sender authentication for email, which would enable some sending charge to emails too. This would have a major impact as we have seen for instance in cellular business where spam short messaging has not happened in a large scale. The cost of SMS to the sender is a prohibiting factor. There are activities ongoing in this area both in Internet Community where Anti Spam Research Group (ARSG), a daughter group of IETF has been established. [19]. Standardization is today in a quite early phase and it may well be that we need to wait for better email standards for quite some time. And even with Internet standards, it takes a long time before the new standards are all also implemented and deployed.

The fourth element in this fight is legislation in all countries, which should make all spamming illegal. Currently, in some states in USA this is already the case but for instance in the UK Opt-out spamming is forbidden only for consumers. It has been shown that Opt-in is the only acceptable approach to differentiate legal electrical direct marketing from spamming. Dedicated interest groups are trying to drive the legislation in the USA, in EU and elsewhere to tighten the laws against spam. In most of the cases this is a very welcome approach as long as there is enough reasoning not call spam anything that moves in the Internet. [20], [21], [22], [23].

Finally, the education of the consumers is also important. They should be made much better aware of the risks of exposing their email address, responding to any internet surveys and enquiries and especially SCAM. Consumers should also require their service providers to protect them better as part of the service.

7.1 New risk areas for SPAM

Spam is now a serious threat for the use of the Internet. The number of computers connected to the Internet and the number of email addresses are today over 600 million.

Wireless devices have already some time ago reached the milestone of 1 billion devices and access numbers in use. The calling party pay – concept has protected the wireless businesses in most countries but with the converged digital technologies mobile devices will include more and more features which make them fully internet compatible including Multimedia Message Service (MMS) and regular email. Especially using email with wireless devices includes immediately the same risks as with ordinary email. Additionally, if the email address is somehow bundled with the telephone number, it may make it impossible for the end user to escape from a polluted email address – he should change his telephone number at the same time. This is not an acceptable approach. This risk has already materialised in Japan because the wireless messaging in Japan in based on the email paradigm, not the calling party pays concept such as SMS.

There are some markets where operators are using or considering the use of called party pay – concept for MMS. They should be informed quite well about the risks involved. The MMS specification supports both concepts but only "sender pays" is safe from spam. [24]

8 Conclusions

It is obvious that spam has a very important role in Internet email, especially in the USA. Significant businesses are utilising spam in their direct marketing but serious business is moving gradually away from spam and they are starting to use acceptable electrical direct marketing methods, like the Opt-in scheme, to select the receivers much more carefully. This is not only improving the feedback rate and success rate in making business but also reduces significantly the blind bulk email in the Internet.

The most severe harm spam is causing to corporations that have to fight spam in order to keep the business processes running and to keep the focus of the workforce on the business, not on the spam. The economical losses of the corporations may exceed the total market value created using spam as advertising media.

Novel schemes have been developed recently to fight against the spammers, which in the longer run may make the business case for spamming negative. Additional legislation and regulation is needed fast to help the service providers and corporations to fight against spam and especially spam using viruses hijacking the consumer's computers and to limit the spamming now. Legislators have to balance between tight policies and adequate protection for the citizens and also protecting the Internet, to keep it clean and useful for so many good things it can provide to us.

Educating the general public to avoid behaviour that may facilitate spamming is important but as important it is to push the Internet service providers and particularly wireless operators to think carefully about the ways to keep the wireless part of the Internet as clean as it has so far been.

The ultimate target can be no less than to clean the network all the way from all harmful emails, keeping in mind that email which may be unwanted to somebody may be appreciated by somebody else.

References

[1] Hobley Christopher, Just numbers, Number of Internet use, electronic commerce, IT and related figures for the European Community January 2001. http://europa.eu.int/ISPO/ecommerce/documents/Just_numbers.pdf

[2] Internet Engineering Task Force, List of Requests for Comments. http://www.ietf.org/rfc.html

[3] Fallows Deborah, Spam, How It is Hurting Email and Degrading Life on the Internet, October 22 2003. PEW Internet & Americal Life. http://www.pewinternet.org/reports/pdfs/PIP_Spam_Report.pdf

[4] Gauthronet Serge, Drouard Etienne; Ei-toivottu kaupallinen viestintä ja tietosuoja, Yhteenveto tutkimuksen (ETD/99/B5-3000/E/96 tuloksista, Euroopan Yhteisöjen Komissio.

http://europa.eu.int/comm/internal_market/privacy/docs/studies/spamsum_fi.pdf

[5] Home page of Brightmail Incorporated. http://www.brightmail.com/spamstats.html

[6] Home page of MessageLabs Incorporated. http://www.messagelabs.com/viruseye/research/default.asp

[7] Liikanen Erkki, Press Conference, Spam: the Challenge. http://europa.eu.int/comm/commissioners/liikanen/media/slides/spam.pdf

[8] Mahowald Robert, Moser Karen, Holding back the Flood. An IDC White paper. http://www.brightmail.com/pdfs/BMI_ROI_IDC.pdf

[9] Sullivan R., Brunker M. SPAM Wars. MSNBC news. August 2003 http://www.msnbc.com/news/SPAM_front.asp

[10] False Claims in Spam, A report by the FTC's Division of Marketing Practises, April 30 2003. Federal Trade Commission, the USA. http://www.ftc.gov/reports/spam/030429spamreport.pdf

[11] Mangalindan Mylene, For Bulk E-Mailer, Pestering Millions Offers Path to Profit. Wall Street Journal Online, November 11 2002.

[12] 24/7REALMEDIA home page. http://www.247realmedia.com/index.html?navSource=TopNav

[13] Spam by Numbers, Eprivacy report, June 2003. http://www.eprivacygroup.com/pdfs/SpamByTheNumbers.pdf

[14] Malik Dale, Notes from "Economics of spam" Panel. FTC span forum April, May 2003-11-24 http://www.ftc.gov/bcp/workshops/spam/Presentations/malik.pdf

[15] Wood Paul, A Spammer in the works, White paper of MessageLabs http://security.iia.net.au/downloads/aspammerintheworks.pdf

[16] Cobb Stephen, The Economics of SPAM, Feb 2003 http://www.spamsquelcher.com/economics_of_spam.pdf

[17] Pantel Patrick, Lin Dekang, SpamCop, Spam Classification & Organization Program. http://www.isi.edu/~pantel/Download/Papers/aaai98.pdf

[18] Spamhouse Home page http://www.spamhaus.org/sbl/sbl-rationale.html

[19] Anti Spam Research Group (ASRG) Home page http://www.irtf.org/asrg/

[20] Coalition against Unsolicited Commercial Email (CAUCE) http://www.cauce.org/index.phtml

[21] Legislative proposals for a new Regulatory Framework for electronic communications http://europa.eu.int/comm/information_society/policy/framework/index_en.htm

[22] S. 877 – CAN-SPAM Act of 2003, United States of America Senate. http://www.congress.gov/cgi-bin/bdquery/z?d108:s.00877:

[23] Legistaltive notice, Republican Policy Committee, US Senate http://rpc.senate.gov/_files/L43cm102203.pdf

[24] 3rd Generation Partnership Project, Specification for Mobile Multimedia Messaging service, TS 22.140 stage 1. http://www.3gpp.org/ftp/Specs/html-info/22140.htm

[25] Tavilla Michael, DelBianco Steve, Let Email users take Spam off the menu http://www.itaa.org/isec/docs/NetChoiceSpamReport5-17.pdf

[26] Duke University, Duke Law and technology review. The future of wireless spam. http://www.law.duke.edu/journals/dltr/articles/2002dltr0021.html

[27] United Nations Conference on Trade and Development (Untad), E-Commerce and Development report 2003-11-26 http://www.unctad.org/Templates/webflyer.asp?docid=4228&intItemID=1528&lang=1

Discussion

Can we learn something from this excursion to the world of broadband networks and networked applications? The emergence or the re-emergence of peer-to-peer applications once again testifies that predicting the development of new services and the ways people will use networks is difficult and the users again and again tend to surprise the incumbent players in the networking industry. Although, at the moment, the popularity of the peer-to-peer applications is largely based on illegally copied audio and video content, we can see several factors that support the lasting nature of peer-to-peer.

First of all, it follows from the economic theory that economically efficient pricing for information goods under free market conditions favours taxation or flat rates for goods and services. Second, due to Moore's law more and more digital devices with richer and richer features will be manufactured and sold to consumers under mass market conditions that help to drive the prices of these goods down. It is only natural that the users will try to make the best of the digital devices and services that they invest in or pay a flat rate each month. This is a fruitful economic background for the emergence of the new popularity of peer-to-peer. There are also numerous possible uses for the peer-to-peer technology. One is the distribution of digital goods without lots of expensive servers. Fair enough, a web based model for selling digital information goods legally would be quite feasible as well. What is important is that with the broadband networks a more efficient method of distributing digital information goods has emerged. Peer-to-peer seems to have an important role in bringing that distribution channel into use.

The winners from this phenomenon are the manufacturers of the digital goods that are sold to consumers and the software companies that serve the new needs of consumers who can be satisfied by the digital devices with the accompanying software. The broadband network is conveniently used to distribute the software with very low cost. The software sustains its price because it contains a secret and quite a lot of embedded competence that is not easy to copy.

Due to flat rates for network services, on the flip side of the coin, abuse of network resources by selfish people has also emerged. The two currently most annoying phenomena in this category are Spam or unwanted email and viruses. One can claim that the logic behind these phenomena is that due to flat rates it is not economical for network operators to authenticate users reliably. Reliable authentication is not likely to come cheap and there is no apparent revenue stream that would immediately pay for the efforts since the operators' own charging is based on flat rates. Under flat rates and unreliable authentication Spam is possible. A spammer can send bulk unwanted email to million of users counting that one in one or ten thousand emails will bring some revenue. Poor authentication and poor security make it possible for spammers to hack into people's PCs on the net and use them as platforms for sending email or hosting other functions necessary in their business.

The logic behind viruses is similar to Spam. Due to poor authentication and security in general, it is possible to penetrate people's PCs connected to the Net. Such a machine is easy to use to send Spam or distribute the virus further. While the victim pays a flat rate, he or she is not accusing the network operator too much. If the victim would be paying for the volume, the operator would have lots of complaints and disputes about billing. The pressure to solve the problem of network security would be obvious if volume based pricing would be introduced.

Possible solutions of the dilemmas

Based on the above discussion we can see the outlines of a market-oriented solution. First, the operators solve the problem of reliable authentication and start providing reliable security services to their users. To cover the costs, the operators introduce volume based or block pricing. In the latter, a quota of traffic is covered by the flat monthly rate and the rest of the traffic may be broken to volume brackets each of which has its own price tag. Under these conditions, the operators offer reliable distribution services to content providers. The content providers start offering flexible content services with a choice of price points for the users. If a general and reliable method of payment for digital services is developed, the content providers charge directly for their information goods. For goods with prices of several Euros (10€ +) the current methods are good enough, for the less expensive ones new technology is needed. An alternative is also to charge flat rates for the content based on subscriptions. After all, subscription based charging has widely been used by newspapers and periodic journals, so the method is not new to the content industry.

Is the outlined solution feasible? At least it seems that besides the technical problems, it faces other challenges that may be more difficult to overcome. The operator challenge is how can the first operator introduce volume-based prices on the market? The benefits come when a great majority of operators implement the principle, the first face a risk of loosing customers.

The non-technical challenges for the content provider include the mental shift from existing price models and distribution channels to a new distribution channel, new marketing methods, new ways of creating added value and prices that fit into the world. A big issue is also the control over the customer interface. If the customer is hidden behind the operator, the content provider will not be happy.

What are the alternative solutions? The paper by Zheng Yan gives the outlines of a possible approach. In this approach the content is tied to the application for presenting the content and the whole thing is protected by the Orwellian software structure with the help of some dedicated hardware supporting the control over everything the user can do with the device. This solution can hardly be described as market oriented. Rather it is totally driven by the copyrights (monopoly rights of content creators) and it creates a great opportunity for one big company to gain control over the whole thing. The minimum one has to admit is that the regulator would be facing a great big challenge in trying to create a sort of market under the conditions of such a technology. This solution also faces challenges. One is – why would the users buy into this technology? The answer is, it must be given to the users for free and earn the money from the content for which the users are ready to pay.

Future of broadband

First we need to define broadband. The traditional definition promoted by ITU-T is that a broadband network provides at least a 256kbit/s access connection to its users. We should question whether this minimum speed satisfies the needs of the killer applications in the network. The killer applications for broadband networks include

- Digital content for entertainment,
- Rich communication services such as email with large attachments,
- Distribution of digital information using both web and p2p models.

The thing the differentiates broadband networks from earlier networks is the support for transporting video. An early sign of this is that the majority of p2p content is video already today.

Also taking a look at the most advanced market, South-Korea at the time of this writing, may be helpful. By the end of 2002 70% of homes were connected to the Internet with speeds over one Megabit per second. Starting from 2007 plans for South-Korea state that all new customers and new developments will be based on Fiber to the Home. In the meantime, they expect to make use of VDSL and Ethernet connectivity for homes.

From this discussion it seems justified to say that 256 kbit/s is not enough to support the transport of high quality video and should consequently be understood just as a transitionary phase to real broadband networks that provide at least some Megabits per second capacity for the end users (using the newest MPEG4 or H.264 coding, it is possible to transfer a TV – quality video stream in real time on a channel of approximately 1 Mbit/s).

The future of broadband is a multifaceted problem and this collection of articles does not really give grounds to provide a fully blown analysis. What bodes well for the broadband networks is that they have proven to be an efficient distribution channel for digital services and goods. The market economy creates incentives for more efficient solutions to take the place of the older ones. Like cars, trains and airplanes created by the industrial economy replaced the wagons drawn by horses, new methods of information distribution will replace the old ones in the Information Society.

Future of Peer-to-Peer

Let us look at the opportunity for peer-to-peer applications using the example of Finland. Let's figure the basic parameters. There are some half a million PCs connected to ADSL and CATV broadband networks. Let's assume that the average connection speed is 512kbit/s both ways that is widely available for under 50€ per month. An average PC may run a 1GHz processor with may be some 200 Mflops of processing power. Let's assume that the user could devote some 10Gbytes of his disk for a new application.

The important thing to notice is that all this diskspace, computing power and network capacity is sunk cost for the users, it has already been paid for and most of the time it is idle. The Question we should ask is: Is all this hitec machinery completely useless or can something useful be made of it during the time it would otherwise be idle? A company who has a software proposition that will catch the interest of our half a million users, will have harnessed a distributed computer with the following (approximate) paramenters:

- 100 TeraFlops of computing power,
- 5 Petabytes of diskspace,
- 25 Gbit/s access speed that can be used at any time.

The last figure assumes that the backbone capacity is only one tenth of the sum of the access capacities of the users. In two years time those figures can be expected at least to triple or quadruple. Already, today this distributed computer has approximately 50 times more processing power than the most powerful single computer in Finland hosted by CSC Scientific Computing. The diskspace is growing fast because the new machines mostly have anything between 40 and 120 Gbytes of disk or even more.

Let us take a modest example. Modest, in order not to create hype. Let's store all the study guides of all Finnish Universities in a dedicated peer-to-peer application with a nice tree of attributes and a set of keywords that will help a user to find anything that he or she wants and so that the topic of interest is taught in a Finnish University. In the next phase, let's put live content in the same tree and we have a user-friendly Video-on-demand University on the Internet accessible by anyone who has a broadband connection, a reasonable PC with a sound card and our peer-to-peer application yet to be written. The nice thing about this approach is that no large farms of media servers are needed, that changes to the content are easy to control and propagate and content is presented to the user in a consistent structure disregarding things that the user is not interested in like find first the right University and the right department etc. Also, no new computers need to be bought! It will all run on the existing distributed computer outlined above. The taxpayers have already paid for the tuition, so why not do it? The advantages compared to a web model are: no web or media servers are needed, existing investment are maximally leveraged, search of content is integrated with the application resulting in an easier to use conscise user interface and content management can be integrated with the application.

What about Copyrights?

Another thing is clear to me, the current controversy between the information content providers on the one hand and the broadband network operators and users on the other featuring large scale violations of copyrights can not continue far to the future. The problem must be solved one way or the other. In this report we have tried to give food for thought to help to find the solution.