The World of Peer-to-Peer (P2P)/All Chapters

en.wikibooks.org

December 29, 2013

Contents

0.1 Foreword

This book intends to explain to you the overall utilization that P2P (Peer-to-Peer) technologies have in today's world, it goes deeper into as many implementations as it can and compares the benefits, problems even legal implications and changes to social behaviors and economic infrastructures. We explain in detail about the technology and how works and try to bring you a vision on what to expect in the future.

The book is organized into different parts, but as this is a work that is always evolving, things may be missing or just not where they should be, you are free to become a writer and contribute to fix things up...

0.2 Reader Comments

If you have comments about the technical accuracy, content, or organization of this document, please tell us (e.g. by using the "discussion" pages or by email). Be sure to include the section or the part title of the document with your comments and the date of your copy of the book. If you are really convinced of your point, information or correction then become a writer (at Wikibooks) and do it, it can always be rolled back if someone disagrees.

0.3 Guide to Writers

Authors/Contributors should register if intending to make non-anonymous contributions to the book (this will give more value and relevance to your opinions and views on the evolution of the work and enable others to talk to you) and try to follow the structure. If you have major ideas or big changes use the discussion area; as a rule just go with the flow.

Conventions

A set of conventions[1] have been adopted on the creation of this book please read about them before you contribute any content on the book's talk page[2].

1 http://en.wikibooks.org/wiki/The%20World%20of%20Peer-to-Peer%20%28P2P%29%2FConventions
2 http://en.wikibooks.org/wiki/Talk%3AThe%20World%20of%20Peer-to-Peer%20%28P2P%29

The following people are authors to this book:
Panic[a]
You can verify who has contributed to this book by examining the history logs at Wikibooks (http://en.wikibooks.org/).
Acknowledgment is given for using some contents from other works like Wikipedia[b], theinfobox:Peer to Peer[c] and Internet Technologies[d], as from the authors .

a http://en.wikibooks.org/wiki/User%3APanic2k4
b http://en.wikipedia.org/wiki/
c http://www.theinfobox.com/index.php/Peer_to_Peer
d http://en.wikibooks.org/wiki/Internet%20Technologies

3

3 http://en.wikibooks.org/wiki/Category%3A

1 What is P2P ?

Figure 1 This is a diagram of a Peer-to-Peer computer network.

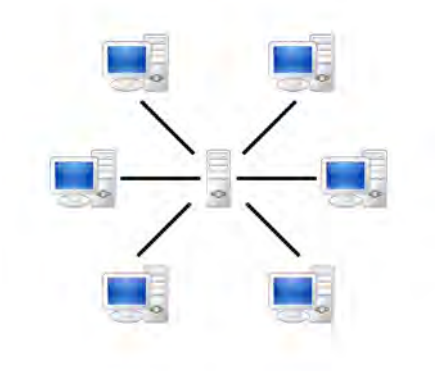

Figure 2 A diagram of a server-based computer network.

Generally, a **peer-to-peer** (or **P2P**) computer network refers to any network that does not have fixed clients and servers, but a number of autonomous *peer* nodes that function as both clients and servers to the other nodes on the network. This model of network arrangement is contrasted with the client-server model (that was unable to scale up to todays necessities). In the P2P model any node should be able to initiate or complete any supported transaction. Peer nodes may differ in local configuration, processing speed, network bandwidth, and storage quantity. This is the basic definition any p2p system.

The term P2P may mean different thing to different people in different contexts. For instance, although the term has been applied to Usenet and IRC in all their incarnations and is even applicable to the network of IP hosts known as the Internet, it is most often used restricted to the networks of peers developed starting in the late 1990s characterized by transmission of data upon the receiver's request instead of the sender's. Such early networks included Gnutella, FastTrack, and the now-defunct Napster which all provide facilities for free (and somewhat anonymous) file transfer between personal computers connected in a dynamic and unreliable way to a network in order to work collectively towards a shared objective.

Even those early Networks did work around the same concept or implementation. In some Networks, such as Napster, OpenNap or IRC, the client-server structure is used for some tasks (e.g. searching) and a peer-to-peer structure for others, and even that is not consistent in each. Networks such as Gnutella or Freenet, use a peer-to-peer structure for all purposes and are sometimes referred to as true peer-to-peer networks, even though some of the last evolution are now making them into a hybrid approach were each peer is not equal in its functions.

When the term peer-to-peer was used to describe the Napster network, it implied that the peer protocol nature was important, but in reality the great achievement of Napster was the empowerment of the peers (ie, the fringes of the network). The peer protocol was just a common way to achieve this.

So the best approach will be to define peer-to-peer, not as a set of strict definitions but to extend it to a definition of a technical/social/cultural movement, that attempts to provide a decentralized, dynamic and self regulated structure (in direct opposition to the old model o central control or server-client model, that failed to scale up to today's expectations), with the objective of providing content and services. In this way a computer programs/protocol that attempts to escape the need to use a central servers/repository and aims to empower or provide a similar level of service/access to a collection of similar computers can be referred to as being a P2P implementation, and it will be in fact enabling everyone to be a creator/provider, not only a consumer. Every P2P system is by definition self feeding, the more participants it has the better it will satisfy it's objectives.

1.1 From a Computer Science Perspective

Technically, a true peer-to-peer application must implement only peering protocols that do not recognize the concepts of "server" and "client". Such *pure* peer applications and networks are rare. Most networks and applications described as peer-to-peer actually contain or rely on some non-peer elements, such as DNS. Also, real world applications often use multiple protocols and act as client, server, and peer simultaneously, or over time.

P2P under a computer science perspective creates new interesting fields for research not on to the not so recent switch of roles on the networks components, but due to unforeseen benefits and resource optimizations it enables, on network efficiency and stability.

Peer-to-peer systems and applications have attracted a great deal of attention from computer science research; some prominent research projects include the Chord lookup service,

the PAST storage utility, and the CoopNet content distribution system (see below for external links related to these projects).

It is also important to notice that the computer is primarily a information devices, whose primary function is to copy data from location to location, even more than performing other types of computations. This makes digital duplication something intrinsic to the normal function of any computer it is impossible to realize the goal of general purpose open computing with any type of copy protection. Enforcement of copyright in the digital era should not be seen as a technical issue but a new reality that society needs to adapt to.

1.1.1 Distributed Systems

Distributed systems[1] are becoming key components of IT companies for data centric computing. A general example of these systems is the Google infrastructure[2] or any similar system. Today most of the evolution of these systems if being focused on how to analyze and improve performance. A P2P system is also a distributed systems and share, depending on the implementation, the characteristics and problems of distributed systems (error/failure detection, aligning machine time, etc...).

Ganglia

Ganglia is a scalable distributed monitoring system for high-performance computing systems such as clusters and Grids. It is based on a hierarchical design targeted at federations of clusters. It leverages widely used technologies such as XML for data representation, XDR for compact, portable data transport, and RRDtool for data storage and visualization. It uses carefully engineered data structures and algorithms to achieve very low per-node overheads and high concurrency.

Ganglia has been ported to an extensive set of operating systems and processor architectures, and is currently in use on thousands of clusters around the world. It has been used to link clusters across university campuses and around the world and can scale to handle clusters with 2000 nodes. (`http://ganglia.info/`)

1.1.2 Distributed Computation

The basic premise behind distributed computation is to spread computational tasks between several machines distributed in space, most of the new projects focus on harnessing the idle processing power of "personal" distributed machines, the normal home user PC. This current trends is an exciting technology area that has to do with a sub set of distributed systems (client/server communication, protocols, server design, databases, and testing).

This new implementation of an old concept has it's roots in the realization that there is now a staggering number of computers in our homes that are vastly underutilized, not only home

1 `http://en.wikipedia.org/wiki/Distributed%20computing`
2 `http://en.wikipedia.org/wiki/Google%23Platform`

computers but there are few businesses that utilizes their computers the full 24 hours of any day. In fact seemingly active computers can be using only a small part of it processing power. Using a word processing, email, and web browsing, require very few CPU resources. So the "new" concept is to tap on this underutilized resource (CPU cycles) that can surpass several supercomputers at substantially lower costs since machines that individually owned and operated by the general public.

SETI@Home

One of the most famous distributed computation[3] project, , hosted by the Space Sciences Laboratory[4], at the University of California, Berkeley[5], in the United States[6]. *SETI* is an acronym for the Search for Extra-Terrestrial Intelligence[7]. SETI@home[8] was released to the public on May 17, 1999.

In average it used hundreds of thousands of home Internet[9]-connected computers in the search for extraterrestrial intelligence. The whole point of the programs is to run your free CPU cycles when it would be otherwise idle, the original project is now deprecated to be included into BOIC[10].

BOINC

BOINC has been developed by a team based at the Space Sciences Laboratory[11] at the University of California, Berkeley[12] led by David Anderson[13], who also leads SETI@home.

Boinc stands for Berkeley Open Infrastructure for Network Computing, a non-commercial (free[14]/w:open source software[15]), released under the LGPL[16], middleware[17] system for volunteer computing[18], originally developed to support the SETI@home[19] project and still hosted at (`http://boinc.berkeley.edu/`), but intended to be useful for other applications in areas as diverse as mathematics, medicine, molecular biology, climatology, and astrophysics. an open-source software platform for computing using volunteered resources that extends the original concept and lets you donate computing power to other scientific research projects such as:

3 http://en.wikipedia.org/wiki/distributed%20computing
4 http://en.wikipedia.org/wiki/Space%20Sciences%20Laboratory
5 http://en.wikipedia.org/wiki/University%20of%20California%2C%20Berkeley
6 http://en.wikipedia.org/wiki/United%20States
7 http://en.wikipedia.org/wiki/SETI
8 http://en.wikipedia.org/wiki/SETI%40home
9 http://en.wikipedia.org/wiki/Internet
10 http://en.wikipedia.org/wiki/Berkeley%20Open%20Infrastructure%20for%20Network%20Computing
11 http://en.wikipedia.org/wiki/Space%20Sciences%20Laboratory
12 http://en.wikipedia.org/wiki/University%20of%20California%2C%20Berkeley
13 http://en.wikipedia.org/wiki/David%20P.%20Anderson
14 http://en.wikipedia.org/wiki/free%20software
15 http://en.wikipedia.org/wiki/open%20source%20software
16 http://en.wikipedia.org/wiki/GNU%20Lesser%20General%20Public%20License
17 http://en.wikipedia.org/wiki/middleware
18 http://en.wikipedia.org/wiki/volunteer%20computing
19 http://en.wikipedia.org/wiki/SETI%40home

- Climateprediction.net: study climate change.
- Einstein@home: search for gravitational signals emitted by pulsars.
- LHC@home: improve the design of the CERN LHC particle accelerator.
- Predictor@home: investigate protein-related diseases.
- Rosetta@home: help researchers develop cures for human diseases.
- SETI@home: Look for radio evidence of extraterrestrial life.
- Folding@Home (`http://www.stanford.edu/group/pandegroup/folding/`): to understand protein folding, misfolding, and related diseases.
- Cell Computing biomedical research. (Japanese; requires nonstandard client software)
- World Community Grid: advance our knowledge of human disease. (Requires 5.2.1 or greater)

As a "quasi-supercomputing" platform, BOINC has over 435,000 active computers (hosts) worldwide. BOINC is funded by the National Science Foundation[20] through awards SCI/0221529, SCI/0438443, and SCI/0506411.

It is also used for commercial usages, as there are some private companies that are beginning to use the platform to assist in their own research. The framework is supported by various operating systems: Windows[21] (XP/2K/2003/NT/98/ME), Unix[22] (GNU/Linux[23], FreeBSD[24]) and Mac OS X[25].

World Community Grid (WCG)

Created by IBM, World Community Grid (`http://www.worldcommunitygrid.org/`) is similar to the above systems. Fourteen IBM servers serve as "command central" for WCG. When they receive a research assignment from an organization, they will scour it for security bugs, parse it into data units, encrypt them, run them through a scheduler and dispatch them out in triplicate to the army of volunteer PCs.

To be a volunteer one only needs to download a free, small software agent (similar to a screensaver).

Projects get selected based on the potential to benefit from WCG technology and address humanitarian concerns, and chosen by an independent, external board of philanthropists, scientists and officials.

The software is OpenSource (LGPL), C/C++ and wxWidgets[26] and is available for Windows, Mac, or Linux.

20 `http://en.wikipedia.org/wiki/National%20Science%20Foundation`
21 `http://en.wikipedia.org/wiki/Windows`
22 `http://en.wikipedia.org/wiki/Unix`
23 `http://en.wikipedia.org/wiki/GNU%2FLinux`
24 `http://en.wikipedia.org/wiki/FreeBSD`
25 `http://en.wikipedia.org/wiki/Mac%20OS%20X`
26 `http://en.wikipedia.org/wiki/wxWidgets`

Grid Networks

Grids first emerged in the use of supercomputers in the U.S. , as scientists and engineers sought access to scarce high-performance computing resources that were concentrated at a few sites.

Open Science Grid
The Open Science Grid (`http://www.opensciencegrid.org/`) was built and is operated by the OSG Consortium, it is a U.S. grid computing infrastructure that supports scientific computing via an open collaboration of science researchers and software developers from universities and national laboratories, storage and network providers.

Globus Alliance
The Globus Alliance (`http://www.globus.org/`) is a community of organizations and individuals developing fundamental technologies behind the "Grid," which lets people share computing power, databases, instruments, and other on-line tools securely across corporate, institutional, and geographic boundaries without sacrificing local autonomy.

The Globus Alliance also provides the Globus Toolkit[27], an open source software toolkit used for building robust, secure, grid systems (peer-to-peer distributed computing on supercomputers, clusters, and other high-performance systems) and applications. A Wiki is available to the Globus developer community (`http://dev.globus.org/wiki/Welcome`).

High Throughput Computing (HTC)

As some scientists try extract more floating point operation per second (FLOPS) or minute from their computing environment, others concentrate on the same goal for larger time scales, like months or years, we refer these environments as High Performance Computing (HPC) environments.

The term HTC was coined in a seminar at the NASA Goddard Flight Center in July of 1996 as a distinction between High Performance Computing (HPC) and High Throughput Computing (HTC).

HTC focus is on the processing power and not on the network, but the systems can also be created over a network and so be seen as a Grid network optimized for processing power.

Condor Project
The goal of the Condor Project (`http://www.cs.wisc.edu/condor/`) is to develop, implement, deploy, and evaluate mechanisms and policies that support High Throughput Computing (HTC) on large collections of distributively owned computing resources. Guided by both the technological and sociological challenges of such a computing environment, the Condor Team has been building software tools that enable scientists and engineers to increase their computing throughput.

27 `http://www.globus.org/toolkit/about.html`

IBM Grid Computing

IBM among other big fishes in the IT pond, spends some resources investigating Grid Computing. Their attempts around grid computing are listed in on the projects portal page (`http://www.ibm.com/grid`). All seem to attempt to leverage the enterprise position on server machines to provide grid services to costumers. The most active project is Grid Medical Archive Solution[28] a scalable virtual storage solution for healthcare, research and pharmaceutical clients.

1.1.3 Content distribution/Hosting

The traditional method of distributing large files is to put them on a central server. The server and the client can then share the content across the network using agreed upon protocols (from HTTP, FTP to an infinite number of variations) when using IP connections the data can be sent over TCP or UDP connection or a mix of the two, this all depends mostly on the requirements on the service, machines, network and many security considerations.

The advantages regarding optimization of speed, availability and consistency of service in regards to optimal localization is nothing unheard off. Akamai Technologies[29] and Limelight Networks[30] among other similar solutions have attempted to commercially address this issue, even Google has distributed the location of its data centers to increase the response of its services. This has addressed the requirement of large content and service distribution but is not is not a fully decentralization of the control structure.

P2P evolved in to solve a distinct problem, that central servers do not scale well. Bandwidth, space and CPU constitute are a point of failure, that can easily bring the function of a system to an end, as any centralization of services.

Note:
One simple example on how centralization is problematic is exemplified by the Denial-of-service attack[a], that generally consists of the efforts of one or more machines to temporarily or indefinitely interrupt or suspend services of a server connected to the Internet, generally by overloading it.

a `http://en.wikipedia.org/wiki/Denial-of-service%20attack`

1.1.4 Conferences and papers

P2P is not, yet, a well established field of research or even a computer science specific field. P2P technology covers too many subjects, that it is yet hard to restrict all interactions as a field on itself. As a result much relevant information is hard to find.

Conferences

28 `http://www.ibm.com/linux/grid/medical_archive_solution.shtml`
29 `http://en.wikipedia.org/wiki/Akamai%20Technologies`
30 `http://en.wikipedia.org/wiki/Limelight%20Networks`

For conferences one of the locations that has update information from a non-commercial and platform agnostic viewpoints is the list provided by the project GNUnet project[31] at `https://gnunet.org/conferences` .

Papers

1.2 From a Economics Perspective

For a P2P system to be viable there must be a one to one equal share of work between peers, the goal should be a balance between consumption and production of resources with the goal of maintaining a singe class of participant on the network, sharing the same set of responsibilities. Most P2P systems have a hard time creating incentives for users to produce (contribute), and end up generating a pyramidal (or multiples, tree like) scheme, as users interact within making the systems dependent on the network effect[32] created. The more users the system has, the more attractive it is (and the more value it has) as any system that depends on the network effect, it's success is based on compatibility and conformity issues.

"in many cases, in fact, the principal reason for which fair use is granted is the extremely high transactions costs that copyright enforcement would require."
—In Richard Watt's book Copyright and Economic Theory

The digital revolution has created a wave of changes some are yet to be fully understood. One of the most important in regards to economics, and that commercial goods producers are fully aware is the dilution of value due to the increasing accumulation and durability of older creations. Digital media has made not only old creations more durable but also more easily accessible and visible and cheap to accumulate.

Even if rarely anyone defends works on the public domain, those works continue to move consumers from the need to acquire new one. It fallows that this makes our common cultural historic records a prime target for the dark forces that arise from basic economic interests.

1.2.1 The popularization of the production and distribution of Cultural goods

"[Software are among the] things which can be copied infinitely over and over again, without any further costs."
—Eben Moglen[33], 2006

31 `http://en.wikipedia.org/wiki/GNUnet`
32 `http://en.wikipedia.org/wiki/network%20effect`
33 `http://en.wikipedia.org/wiki/Eben%20Moglen`

P2P radically shifts the economics of distribution and business models dealing with intangible cultural goods (intellectual property[34]). Since most content is virtual, made only of information. This information can be any type of non material object that is made from ideas (text, multimedia, etc.). In this way content is also the myriad ways ideas can be expressed. It may consist of music, movies, books or any one single aspect, or combination of each.

Music

The digital revolution has forced to music industry to reshape itself in various ways. From the promotion, to distribution passing.

Radio Radio had been for a long time the way that the industry managed demand for its products. It was not a question of quality but of product "visibility" and a easy way to generate revenue from royalties[35].

The advent of the transistor and the walk-man should have clued-in the industry to the changes to come. Even here the simple digitalization and the possibility of moving radio from the airwaves to the Internet has caused much pain in the industry and eroded considerably how it had managed to shape demand.

From Tape to MP3 Another front of attack came from the very media where the content was sold. Something that should also been foresaw since the advent of the 8-track tape[36], that culminated in compact Cassette[37] was another revolutionary change to the business model, as revolutionary to the music industry as it was to the video. Adaptation led to the CD and mixed content offerings. In fact it seems that all pressure and incentives to change were being dictated by the consumers and the industry economic effort to reduce production costs but those that had decision power over the industry continued to remain blind to the technological changes they were fostering themselves.

Here the move to the digital not only permitted even easier reproduction but ultimately made the product completely virtual and independent from the media it was sold on. The walk-man, evolved into the portable Cd-player and ultimately died silently as solid memory and portable players took over the market.

Internet

UNKNOWN TEMPLATE Sidenote

UNKNOWN TEMPLATE YouTube link

XZUSn7I-zNoDeath of ACTA music video

34 http://en.wikibooks.org/wiki/Strategy%20for%20Information%20Markets%2FIntellectual%20Property

35 http://en.wikipedia.org/wiki/Royalties

36 http://en.wikipedia.org/wiki/8-track%20tape

37 http://en.wikipedia.org/wiki/Compact%20Cassette

Dan Bull[38]'s D.O.A.C.T.A (Death of ACTA) being ACTA the Anti-Counterfeiting Trade Agreement, in it the creator explains concisely how independent music creators see this continued call for intellectual property control policies.

The need for content intermediaries continues its rapid decline. Most intermediaries do not add much value to the product besides being able to provide better marketing orientation and general business knowhow to the content producers.

The time where volume would permit record companies to offer better production facilities is over, as the price for producing an audio work is now accessible to all, even if in physical form. Intermediaries in fact are becoming to costly for the perks they can still provide. They create unnecessary barriers between the producers and consumers.

In todays interconnected world the distribution channels are so diversified that creating artificial control schemes for digital distribution (physical or virtual) will only degrade the level of satisfaction of consumers without increasing product value but incrementing the costs to the sanctioned distributors.

If costumers are faced with a product with DRM, then unauthorized copies if made publicly available, will create a competing product without limitations, thus creating a better product with a better price tag. In fact the use of DRM creates differentiation and promotes the creation of a parallel markets (if one can call it that because most offerings are gratis, but multiple DRM schemes would fragment the market in the same way), this results from the consumers wishes are not being satisfied by the primary offer or by simply enabling more choices.

Today radio, TV and the press as a publicity vehicle is becoming increasingly infective in relation to the interactive media that the Internet permits and it can be utilized as a direct distribution channels. More and more artist are becoming aware of the advantages of controlling the copyright of their productions and taking the responsibility of distributing their own works, this has also increased the level of communication with the consumer.

This has become quickly evident in the music industry, mostly because the medium has always been extremely volatile and consumers have had a great number of ways of utilizing the content, reducing the freedom of movement for the content have always been attempted and failed, the same is becoming true for video and with time even books will have to deal with this new reality, as is now seen with the written press. As the medium for the content becomes ubiquitous, cheap and acceptable to consumers, producers will have to adapt.

Video

Movies

TV

Recently some television networks are rethinking their approach to audiences, this has resulted from the level acceptance and interest that DVD show collections were having and several online attempts to improve distribution. Since now anyone can easily illegally download their favorite shows, a problem similar to the fragmentation of the distribution channels

38 http://en.wikipedia.org/wiki/Dan%20Bull

as seen in the music recording industry with the rise of alternative delivery technologies will have a similar result if television industry fails adapt and fill the audiences expectations of quick and easy accessibility to new fresh content.

ISPs

ISPs have been shaping/throttling P2P traffic, especially the more popular networks for years, resulting on an ongoing cat and mouse game between ISPs and P2P developers. In the US the network neutrality discussion and recently the evidence of this actions by ISPs against P2P traffic has turned this matter into a political issue.

In November 2007, Vuze, creators of Azureus (a Bittorrent application), petitioned the FCC, resulting in a FCC hearing held in December 2007. One of the issues raised there, was the level of data available on BitTorrent throttling. This lead to a statement by the General Counsel at Vuze, Jay Monahan; "We created a simple software "plug-in" that works with your Vuze application to gather information about potential interference with your Internet traffic."

This plugin[39] has been gathering more hard data on the actions of ISPs, resulting in a growing list of ISPs that interfere with P2P protocols is maintained on the Azureus WIKI (`http://www.azureuswiki.com/index.php/Bad_ISPs`).

1.3 From a Sociological Perspective

From person to person or user to user, a new world is being born in that all are at the same time producers and consumers. The information will be free since the costs of distribution will continue to fall and the power for creative participation is at anyones hands.

"There's a war out there, old friend. A world war. And it's not about who's got the most bullets. It's about who controls the information. What we see and hear, how we work, what we think... it's all about the information! "
—from a dialog in Sneakers (film)[40]

1.3.1 Is it morally wrong?

As discussed previously there is no common ground to answer this question, views differ wildly, even states degrees with the interpretation or legality of restricting/implementing intellectual property rights.

For every action there is a reaction

It is today an evidence that there is a social movement against what is generally perceived as the corruption of copyright over public goods, that is, legally a minority is attempting

39 `http://azureus.sourceforge.net/plugin_details.php?plugin=aznetmon`
40 `http://en.wikipedia.org/wiki/Sneakers%20%28film%29`

to impose extensions and reductions of liberties to defend economical interests of mostly sizable international corporations that in it's vast majority aren't even the direct creators of the goods. In this particular case virtual goods, mostly digital that have a approaching 0 cost of replication and aren't eroded by time or use.

1.3.2 DRM (Digital Rights Management)

"Once again, content owners have successfully promoted their own narrow financial interests over the broader public interest in preserving consumer access to literary, scientific, and other works"
—Peter Jaszi, Professor at American University's Washington College of Law.

When we talk about DRM it useful to keep in mind that the rights that are being "managed" are completely distinct from simple intellectual rights that were granted protection on non-digital media. Since the level of control permitted can be extreme, for those that respect the DRM, sometimes the rights that are removed from the consumer are simply freedom that existed in past media, for example the freedom to lend. It has gone to a point that the concept of buying a good has been slowly changing to renting, in a way that you ultimately do not own or have full control over what you paid for.

Note:
The issues regarding DRM has become of such a magnitude that as this type of technology becomes more abusive of consumers rights that its surreptitious implications that due to the lack of consumer's awareness and education efforts have began to appear to provide labels and indicators that permit a more educated selection of what is consumed. The Free Software Foundation[a] via its defectivebydesign.org[b] campaign started in 2008 a promoting the adoption of a distinctive label indicating DRM-Free content/media (`http://www.defectivebydesign.org/drm-free`).

a `http://en.wikipedia.org/wiki/Free%20Software%20Foundation`
b `http://www.defectivebydesign.org`

`http://`

A more unified marker for DRM-free files that also educates downloaders about DRM is a powerful way to increase the value of being DRM-free. People looking for ebooks in places like Amazon often have trouble figuring out which ebooks have DRM and which don't because Amazon does not advertise that information. This label is a step toward solving that problem, making it easy for people who oppose DRM to find like-minded artists, authors, and publishers to support.

In late 2005, market-based rationales influenced Sony BMG's deployment of DRM systems on millions of Compact Discs that threatened the security of its customers computers and compromised the integrity of the information infrastructure more broadly. This became known as the Sony BMG Rootkit debacle (see the paper Mulligan, Deirdre and Perzanowski,

Aaron K., "The Magnificence of the Disaster: Reconstructing the Sony BMG Rootkit[41], for detailed information).

In February 2007, Steve Jobs, wrote an open letter addressing DRM since it was impacting Apples business on the iTunes/iPod store (`http://www.apple.com/hotnews/thoughtsonmusic/`).

On a presentation made by David Hughes of the RIAA at Arizona State University (2007), David Hughes, senior vice president of technology for the RIAA, dubbed the spiritual leader of Apple Steve Jobs as a "hypocrite" over his attitude to DRM on iTunes. "While Steve has been banging on about the music companies dropping DRM he has been unwilling to sell his Pixar movies through iTunes without DRM and DVDs without CSS encryption."

a danger for historical records

1.3.3 P2P United

A now disbanded organization formed by six of the biggest P2P groups (those behind eDonkey, Grokster, Morpheus, Blubster, Limewire and BearShare), with Adam Eisgrau as executive director. It was started in mid-July 2003 to provide a way to lobby for the P2P on U.S. Congress and WIPO, the UN organization that administers intellectual property treaties since the file-sharing industry (as an industry) had no identifiable name and face in Washington or in the media.

This attempt was a bust and since then most of the members of the group has lost court cases or have settled and closed operations.

1.3.4 Peer-to-Peer working group

The Peer-to-Peer WG[42] (P2Pwg).

A great article about problems with the creation of the working group is available at (www.openp2p.com) by Tim O'Reilly 10/13/2000 is available (`http://www.openp2p.com/pub/a/p2p/2000/10/13/working_grp.html`).

1.3.5 P2P in non technological fields

There are also several movements attempting to establish how to apply the concept of P2P to non technological fields like politics, economics, ecology etc...

One of such attempts is the The Foundation for P2P Alternatives (`http://p2pfoundation.net`), that function as a clearinghouse for such open/free, participatory/p2p and commons-oriented initiatives and aims to be a pluralist network to document, research, and promote peer to peer alternatives.

41 `http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1072229`
42 `http://p2p.internet2.edu/`

1.4 From a Legal Perspective

The most commonly shared files on such networks are mp3 files of popular music and DivX movie files. This has led many observers, including most media companies and some peer-to-peer advocates, to conclude that these networks pose grave threats to the business models of established media companies. Consequently, peer-to-peer networks have been targeted by industry trade organizations such as the RIAA and MPAA as a potential threat. The Napster service was shut down by an RIAA lawsuit; both groups the RIAA and MPAA spend large amounts of money attempting to lobby lawmakers for legal restrictions. The most extreme manifestation of these efforts to date (as of January, 2003) has been a bill introduced by California Representative Berman, which would grant copyright holders the legal right to break into computer systems believed to be illegally distributing copyrighted material, and to subvert the operation of peer-to-peer networks. The bill was defeated in committee in 2002, but Rep. Berman has indicated that he will reintroduce it during the 2003 sessions.

As attacks from Media companies expand the networks have seemed to adapt at a quick pace and have become technologically more difficult to dismantle. This has caused the users of such systems to become targets . Some have predicted that open networks may give way to closed, encrypted ones where the identity of the sharing party is not known by the requesting party. Other trends towards immunity from media companies seem to be in wireless adhoc networks where each device is connected in a true peer-to-peer sense to those in the immediate vicinity.

While historically P2P file sharing has been used to illegally distribute copyrighted materials (like music, movies, and software), future P2P technologies will certainly evolve and be used to improve the legal distribution of materials.

As it should be obvious by now the problem P2P technologies create to the owner of the content, to the control of the distribution channels and to the limitation of users (consumers) rights is huge, the technology is making holes in the standard ideology that controls the relations between producers and consumers some new models have been proposed (see for example Towards solutions to “the p2p problem” - `http://groups.sims.berkeley.edu/pam-p2p/`).

In 2007 a handful of the wealthiest countries (United States, the European Commission, Japan, Switzerland, Australia, New Zealand, South Korea, Canada, and Mexico) started secretive negotiations toward a treaty-making process to create a new global standard for intellectual property rights enforcement, the Anti-Counterfeiting Trade Agreement (ACTA)[43] initially due to be adopted at the 34th G8 summit in July 2008, has now hoped to be concluded in 2010.

It has been argued that the main purpose of the treaty is to provide safe harbor for service providers so that they may not hesitate to provide information about infringers; this may be used, for instance, to quickly identify and stop infringers once their identities are confirmed by their providers.

43 `http://en.wikipedia.org/wiki/Anti-Counterfeiting%20Trade%20Agreement`

Similarly, it provides for criminalization of copyright infringement, granting law enforcement the powers to perform criminal investigation, arrests and pursue criminal citations or prosecution of suspects who may have infringed on copyright.

More pressingly, being an international treaty, it allows for these provisions—usually administered through public legislation and subject to judiciary oversight—to be pushed through via closed negotiations among members of the executive bodies of the signatories, and once it is ratified, using trade incentives and the like to persuade other nations to adopt its terms without much scope for negotiation.

1.4.1 Is it Illegal?

Peer-to-Peer in itself in nothing particularly new. We can say that an FTP transfer or any other one on one transfer is P2P, like an IRC user sending a DCC file to another, or even email, the only thing that can be illegal is the use one can give to a particular tool.

"There's no way to rule innocent men. The only power government has is the power to crack down on criminals. When there aren't enough criminals, one makes them. One declares so many things to be a crime that it becomes impossible for men to live without breaking laws."
—Ayn Rand[44]

Legal uses of P2P include distributing open or public content, like movies, software distributions (Linux, updates) and even Wikipedia DVDs are found on P2P Networks[45]. It can also be used to bypass censorship, like for instance the way Michael Moore's film 'Sicko' leaked via P2P[46] or as publicity machine to promote products and ideas or even used as a market annalists tool.

However trading copyrighted information without permission is illegal in most countries. You are free to distribute your favorite Linux distribution, videos or pictures you have taken yourself, MP3 files of a local band that gave you permission to post their songs online, maybe even a copy an open source software or book. The view of legality lies foremost on cultural and moral ground and in a globally networked world there is no fixed line you should avoid crossing, one thing is certain most people don't produce restricted content, most view their creations as giving to the global community, so it's mathematically evident that a minority is "protected" by the restrictions imposed on the use and free flow of ideas, concepts and culture in general.

P2P as we will see is not only about files sharing, it is more generally about content/services distribution.

44 http://en.wikipedia.org/wiki/Ayn%20Rand
45 http://en.wikinews.org/wiki/Special:Search?search=p2p&go=Go
46 http://en.wikinews.org/wiki/Michael_Moore%27s_new_film_%27Sicko%27_leaked_via_P2P

Figure 3 Sharing is not theft and theft is not the same as piracy, this is true under any law.

is sharing theft? and is theft piracy? surly not...

Sharing contents that you have no right to is not theft. It has never been theft anywhere in the world. Anyone who says it is theft is wrong. Sharing content that you do not own (or have the rights to distribute) is copyright infringement. Duplicating a digital good does not reduce the value of the original good, nor does it signify a subtraction of the same from the owner. On the other hand, making that same digital good available to others without a license may have a well understood effect of augmenting the visibility of said product, resulting in free advertisement and discussion about the product, this generally results in the increase of the demand for it, this has been validated in tests done with digital books, music and video releases.

Using the term "piracy" to describe the copyright infringement is a metaphoric heuristic, a public relations stunt from lobbies of big corporations that represent copyright holders or hold the copyright over commercializable cultural goods, as a way to mislead the public and legislators. Leading to practices that directly damage society and culture (see Sonny Bono Copyright Term Extension Act or the Mickey Mouse Protection Act[47]).

The legal battles we are now accustomed to hear about, deal mostly on control and to a lesser degree in rights preservation. Control over the way distribution is archived (who gets what in what way) results in creating artificial scarcity. This deals with money, as there is

47 http://en.wikipedia.org/wiki/Copyright%20Term%20Extension%20Act

added value to controlling and restricting access due to format and limitations in time and space.

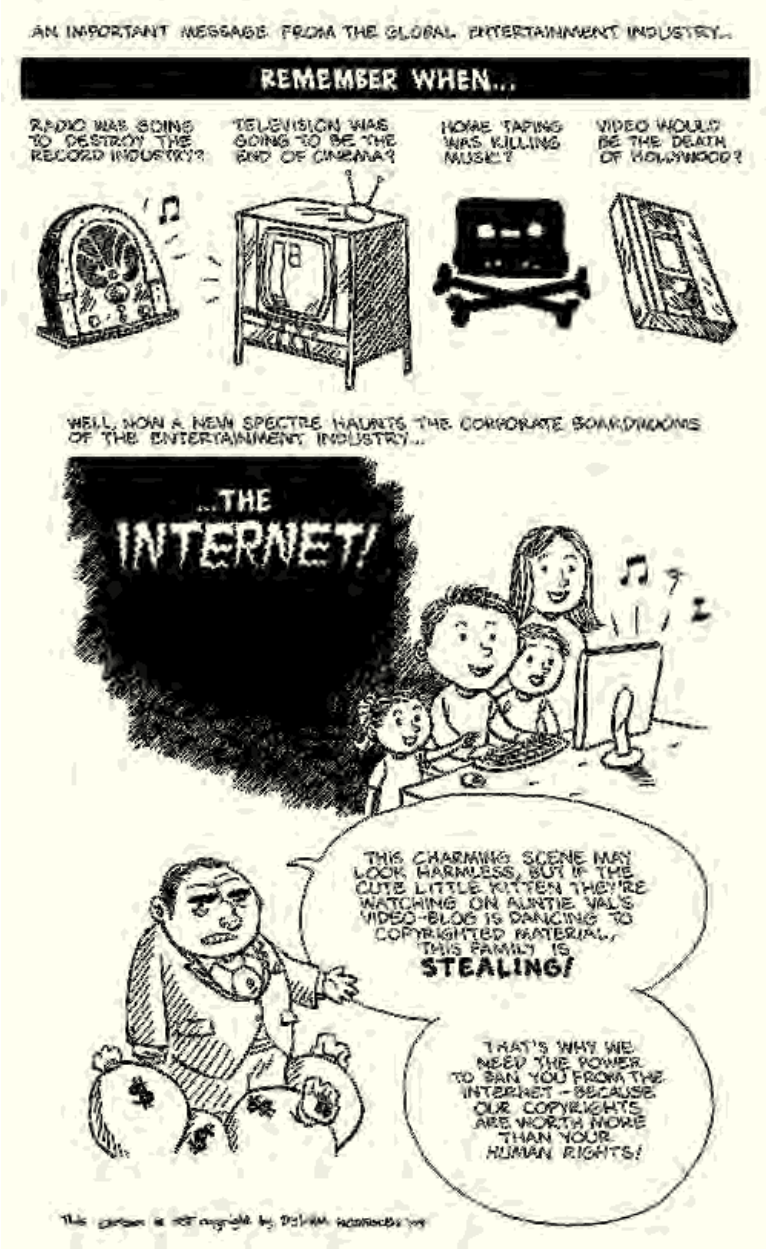

Figure 4 Cartoon about free culture, intellectual property and Internet Piracy. Found on The Pirate Bay in late February / early March '09.

1.4.2 World Intellectual Property Organization (WIPO)

The World Intellectual Property Organization[48] is one of the specialized agencies of the United Nations. WIPO was created in 1967 with the stated purpose "to encourage creative activity, [and] to promote the protection of intellectual property throughout the world". The convention establishing the World Intellectual Property Organization[49], was signed at Stockholm on July 14, 1967.

1.4.3 European Union

Figure 5 An Italian manifest saying "to share is not steal", referring to P2P legal status in Italy.

In August 2007, the Music industry was rebuffed in Europe on file-sharing identifications, as a court in Offenburg, Germany refused to order ISPs to identify subscribers when asked to by Music Industry who suspected specified accounts were being used for copyright-infringing file-sharing, the refusal was based in the courts understanding that ordering the ISPs to handover the details would be "disproportionate", since the Music Industry representatives had not adequately explained how the actions of the subscribers would constitute "criminally relevant damage" that could be a basis to request access to the data.

This was not an insulate incident in Germany, as also in 2007, Celle chief prosecutor's office used the justification that substantial damage had not been shown to refuse the data request, and does follows the opinion of a European Court of Justice (ECJ) Advocate-General, Juliane Kokott who had published an advice two weeks earlier, backing this stance, as it

48 http://en.wikipedia.org/wiki/WIPO

49 http://en.wikipedia.org/wiki/Convention%20Establishing%20the%20World%20Intellectual%20Property%20Organization

states that countries whose law restricted the handing over of identifying data to criminal cases were compliant with EU Directives. The produced advice was directed to a Spanish case in which a copyright holders' group wanted subscriber details from ISP Telefonica. The ECJ isn't obliged to follow an Advocate-General's advice, but does so in over three-quarters of cases.

In most European countries, copyright infringement is only a criminal offense when conducted on a commercial scale. This distinction is important because public funds are not directed into investigating and persecuting personal and most often private copyright violations with limited economic impact.

France

On June 12 2007 the Société des Producteurs de Phonogrammes en France (SPPF - `http://www.sppf.com/`), an entity that represents the legal interests and collects copyright revenue in behalf of independent French audio creations, have publicly announced that they had launched a civil action on the Paris Court of First Instance requesting a court order to terminate the distribution and function of Morpheus (published by Streamcast), Azureus and demanding compensation for monetary losses. In 18 September 2007 a similar action was made against Shareaza and in 20 December 2007 the SPPF announced a new action this time against Limewire[50]. All of this legal actions seem to have as a base an amendment done to the national copyright law that stipulates that civil action can taken against software creators/publishers that do not take steps in preventing users from accessing illegal content.

United Kingdom

Federation Against Copyright Theft (FACT)

The FACT[51] is a trade organization in the United Kingdom established to represent the interests of the its members in the film and broadcasting business on copyright and trademark issues. Established in 1983, FACT works with law enforcement agencies on copyright-infringement issues.

FACT has produced several adverts which have appeared at the beginning of videos and DVDs released in the UK, as well as trailers shown before films in cinemas. While operating with the same function that the Motion Picture Association of America (MPAA)[52], FACT has avoided public outcry by focusing most o its actions in targeting serious and organized crime involving copyright-infringement.

In an interesting demonstration of cross-border mutual support between similar business organizations, in 2008, FACT helped the MPAA in a sting operation against streaming links site SurfTheChannel. The MPAA not only participated in the questioning by bringing their own investigators it was allowed to examine the apprehended equipment and managed to find an United States programmer that worked for the site to testify in the legal proceedings. The programmer was not persecuted in the US, but agreed to pay the MPAA the amount he

50 `http://www.sppf.com/telecharger/CP%20SPPF-%20Limewire%20201207.pdf`
51 `http://en.wikipedia.org/wiki/Federation%20Against%20Copyright%20Theft`
52 Chapter 1.4.2 on page 20

made for working at SurfTheChannel (see MPAA Agents Expose Alleged Movie Pirates[53] for details).

Spain

In July 2009 in Barcelona, Spain. Judge Raul N. García Orejudo declared that "P2P networks, as a mere transmission of data between Internet users, do not violate, in principle, any right protected by Intellectual Property Law," when dismissing the Sociedad General de Autores y Editores (SGAE[54]) legal action for the closing of the eD2K link site elrincondejesus.com..

Portugal

In August 2009, the Inspecção-Geral das Actividades Culturais (IGAC[55]) sent a notification to the biggest ISP in Portugal, Portugal Telecom[56] (PT), to remove pages that hosted links to freely by unlicensed downloadable copyrighted material from external pages. This notification came into being after the situation was reported by the Internet Anti-piracy Civic Movement (Movimento Cívico Antipirataria na Internet, MAPINET), this association most prominent members are the Association for the Audiovisual Commerce of Portugal (Associação de Comércio Audiovisual de Portugal, ACAPOR), Portuguese Phonographic Association (Associação Fonográfica Portuguesa, AFP), the Association for the Management and Distribution of Rights (Associação para a Gestão e Distribuição de Direitos, AUDIOGEST), the Federation of Video Editors (Federação de Editores de Videogramas, FEVIP), the Cooperative for the Management of the Artists Rights (Cooperativa de Gestão dos Direitos dos Artistas, GDA), the Association for the Management of Author's Rights(Associação para a Gestão de Direitos de Autor, GEDIPE, part of the The AGICOA Alliance[57]), the Portuguese Society of Authors (Sociedade Portuguesa de Autores, SPA) and some other producers and editors as well as some translators. Since no real illegal content is hosted on servers under the Portugal Telecom control it seems the links aren't a violation on the users terms of use at the identified Blog portal service, and no action has been taken so far. List of the offending 28 pages available in an article in Portuguese[58].

1.4.4 Norway

The Norway's Personal Data Act (PDA) makes it mandatory for ISPs in the country to delete all IP address logs on their customers more than three weeks old as it is considered personal information. This is a huge step forward in personal data protection laws but it also will make the work of "pirate-hunters" more dificult. The Simonsen law firm, is an

53 http://torrentfreak.com/undercover-mpaa-agents-expose-alleged-movie-pirates-120521/
54 http://en.wikipedia.org/wiki/SGAE
55 http://en.wikipedia.org/wiki/Inspec%C3%A7%C3%A3o%20Geral%20das%20Actividades%20Culturais
56 http://en.wikipedia.org/wiki/Portugal%20Telecom
57 http://en.wikipedia.org/wiki/Association%20of%20International%20Collective%20Management%20of%20Audiovisual%20Works
58 http://economico.sapo.pt/noticias/pt-notificada-para-remover-sites-piratas-do-sapo_67408.html

example since it is known by the lawyer Espen Tøndel, figure head on this matters, and for having since 2006 (now terminated), a temporary license from Norway's data protection office to monitor suspected IP addresses without legal supervision.

1.4.5 USA

Under US law "the Betamax decision" (Sony Corp. of America v. Universal City Studios, Inc.[59]), case holds that copying "technologies" are not *inherently* illegal, if substantial non-infringing use can be made of them. This decision, predating the widespread use of the Internet[60] applies to most data networks, including peer-to-peer networks, since legal distribution of some files can be performed. These non-infringing uses include sending open source[61] software, creative commons[62] works and works in the public domain[63]. Other jurisdictions tend to view the situation in somewhat similar ways.

"With the law books filled with a great assortment of crimes, a prosecutor stands a fair chance of finding at least a technical violation of some act on the part of almost anyone. In such a case, it is not a question of discovering the commission of a crime and then looking for the man who has committed it, it is a question of picking the man and then searching the law books, or putting investigators to work, to pin some offense on him."
—Robert H. Jackson[64], 1940

The US is also a signatory of the WIPO treaties, treaties that were partially responsible for the creation and adoption of the Digital Millennium Copyright Act (DMCA).

As stated in US Copyright Law[65], one must be keep in mind the provisions for fair use, licensing, copyright misuses and the statute of limitations.

MGM v. Grokster

`http://`

Recording Industry Association of America (RIAA)

59 `http://en.wikipedia.org/wiki/Sony%20Corp.%20of%20America%20v.%20Universal%20City%20Studios%2C%20Inc.`
60 `http://en.wikipedia.org/wiki/Internet`
61 `http://en.wikipedia.org/wiki/open%20source`
62 `http://en.wikipedia.org/wiki/creative%20commons`
63 `http://en.wikipedia.org/wiki/public%20domain`
64 `http://en.wikipedia.org/wiki/Robert%20H.%20Jackson`
65 `http://en.wikibooks.org/wiki/%3AUS%20Copyright%20Law`

> The RIAA and the labels took an aggressive stance as soon as online music file sharing became popular. They won an early victory in 2001 by shutting down the seminal music-sharing service Napster.The site was an easy target because Napster physically maintained the computer servers where illegal music files, typically in high-fidelity, compressed, download-friendly MP3 format, were stored. [With P2P networks, the files are stored on individual user computers; special software lets consumers "see" the files and download them onto their own hard drives.]
> —Daphne Eviatar, "Record industry, music fans out of tune," The Recorder, August 20, 2003

The Recording Industry Association of America (RIAA) (`http://www.riaa.com/`) is the trade group that represents the U.S. recording industry. The RIAA receives funding from the four of the major music groups EMI, Warner, Sony BMG and Universal and hundreds of small independent labels.

Motion Picture Association of America (MPAA)

The MPAA[66] is an American trade association that represents six biggest Hollywood studios. It was founded in 1922 as the Motion Picture Producers and Distributors of America (MPPDA). It focus in advancing the business interests of its members and administers the MPAA film rating system.

In the early 1980s, the Association opposed the videocassette recorder (VCR) on copyright grounds. In a 1982 congressional hearing, Valenti decried the "savagery and the ravages of this machine" and compared its effect on the film industry and the American public to the Boston Strangler.

The MPAA acts as a lobbies for stricter legislation in regards to copyright safeguards, protection extensions and sanctions, and actively pursues copyright infringement[67], including fighting against sharing copyrighted works via peer-to-peer file-sharing networks, legally and in technologically disruptive ways.

The MPAA has promoted a variety of publicity campaigns designed to increase public awareness about copyright infringement, including *Who Makes Movies?*[68]; *You can click, but you can't hide*[69]; and *You Wouldn't Steal a Car*, a 2004 advertisement appearing before program content on many DVD[70]s.

The MPAA British counterpart is the Federation Against Copyright Theft (FACT)[71].

66 `http://en.wikipedia.org/wiki/Motion%20Picture%20Association%20of%20America`
67 `http://en.wikipedia.org/wiki/copyright%20infringement`
68 `http://en.wikipedia.org/wiki/Who%20Makes%20Movies%3F`
69 `http://en.wikipedia.org/wiki/You%20can%20click%2C%20but%20you%20can%27t%20hide`
70 `http://en.wikipedia.org/wiki/DVD`
71 Chapter 1.4.2 on page 20

1.4.6 Canada

Canada has a levy on blank audio recording media, created on March 19, 1998, by the adoption of the new federal copyright legislation. Canada introduced this levy regarding the private copying of sound recordings, other states that share a similar copyright regime include most of the G-7 and European Union members. In depth information regarding the levy may be found in the Canadian copyright levy on blank audio recording media FAQ (`http://neil.eton.ca/copylevy.shtml`).

With borders and close ties to it neighbor, Canada as historically been less prone to serve corporations interests and has a policy that contrasts in its social aspects with any other country in the American Continent. The reality is that Canada has been highly influenced and even pressured (economically and politically) by its strongest neighbor, the USA, to comply with its legal, social and economic evolution. In recent time (November 2007) the government of Canada has attempted to push for the adoption of a DMCA-modeled copyright law, so to to comply with the WIPO treaties the country signed in 1997 in a similar move to the USA, this has resulted in a popular outcry against the legislation and will probably result in it's alteration. The visibility of this last attempt was due to efforts of Dr. Michael Geist[72], a law professor at the University of Ottawa considered an expert in copyright and the Internet, that was afraid that law would copy the worst aspects of the U.S. Digital Millennium Copyright Act.

1.4.7 Darknets vs Brightnets

Due the refusal to legislate in accordance with the public needs and wants. By adding extensions of copyrights (US, UK) and by actively promoting the promulgating laws similar to the DMCA in other countries a monoculture is created where a virtual monopoly on cultural goods is created, generating something of a cultural imperialism.

This reality promotes the population to move its support for transparent distribution systems (brightnets) to more closed system (darknets), that will increasingly depend on social connections to get into, like the hold speakeasy bars that popped up during the prohibition, legislating against the people once more will prove to be a failure.

A P2P brightnet for content distribution,where no one breaks the law, so no one need hide in the dark, can only be feasible if built around owner-free media/system or by being as heavily controlled, owned media/system, as the old centralized system. This probably around a centralized entity that should guarantee control and manage the content, maybe even requiring the use of some DRM sheme, an overall failed system.

This types of networks were already tested and failed, since content is also information, the need for privacy or lack of it on an open system will always create generate a more layered system that will ultimately degenerate in a darknet to survive legal actions.

72 `http://en.wikipedia.org/wiki/Michael%20Geist`

1.5 Shadow play

In any open society, secrecy, intentional obfuscation of facts and usurpation or suppression of rights should be seen not only as negatively but often as a civilizational back-step. Often illegal or strictly restricted this types of activities are not only possible by states but by actively perused by large corporations, that in some nations (or standard setting organizations) are able to exert a unprecedented control over policy.

In world where the end user intends to have control over their own hardware and software some actions are not intended to see the light of day. This section is dedicated to bring out some of the subjects/actions in an attempt to help the reader to fully appreciate some of the less publicized information that has some kind of baring on the evolution of P2P.

"Truth is the greatest of all national possessions. A state, a people, a system which suppresses the truth or fears to publish it, deserves to collapse."
—Kurt Eisner[73]

Some organizations (or groups with invested interests) still think that it is up to them to think for the masses, in place of just try to push the information out granting the public the ability to make their own informed decisions. There is an clear organized attempt to, in general, hide facts from the public. Information is power.

Attempting to control access and the flow of information is a fool's errand[74]. People have always rebelled against industry-mandated black box solutions and their artificial restrictions, that do not serve any other purpose but the economic interests of those attempting to exert control.

1.5.1 OS

Since P2P (and P2P related technologies) started to pop up and gather momentum, the security of the user on the OS started to be championed, and placed above user freedoms, by imposing choices that disregards the need to proper educate the users, diminishing their ability to make their own choices. This philosophy keeps people in a state of being technologically challenged, and afraid of change. It is even funny to see to what degree, efforts are made to keep these "security enhancements" hidden.

Well not all is lost, some people can't seem to be made to comply with this state of things and some information can be found and actions reversed.

about MS Windows

- *TCPIP.SYS* - `http://www.lvllord.de` fix[75],`http://microsoft.weblogsinc.com/2004/08/12/xp-sp2-could-slowdown-p2p-downloads/` info[76] for Windows XP.

73 `http://en.wikipedia.org/wiki/Kurt%20Eisner`
74 `http://en.wikipedia.org/wiki/Fool%27s%20errand`
75 `http://en.wikibooks.org/wiki/%20fix`
76 `http://en.wikibooks.org/wiki/%20%20info`

- *CrackTcpip.sys* - http://www.mydigitallife.info/2008/01/07/cracktcpipsys-driver-for-vista-sp1-v668-to-patch-tcpipsys-60600117052/ similar solution for Vista SP1[77]
- BadVista.org[78] - An information campaign run by FSF/GNU aiming to stopping Microsoft Windows Vista adoption by promoting free software. What's wrong with Microsoft Windows Vista?[79] (http://badvista.fsf.org/what-s-wrong-with-microsoft-windows-vista) gives an extensive list of problems in the OS related to user loss of freedom and DRM.

1.5.2 ISPs

One of the most important aspects of running an Internet providers (ISPs) is being quick to adapt to customers' demands and changes in demands and uses. This adaptation can be done from two sides, adapting by modifying the network offering and creating new service offerings or by the suppression and degradation of new consumer trends. The latter is easily done if the ISPs is part of a larger media outlet, that can not only influence public opinion but shape legislation.

ISPs are not very pleased with P2P technologies due to the load the bring into to their networks, although they sell their Internet connections as unlimited usage, if people actually take on their offer, ISPs will eventually be unable to cope with the demand at the same price/profit level. The simpler solution would be to match their offer to the use, by increasing their capacity and fulfilling at least the contractual obligations, but some decided to simply throttle (*i.e.* slow down) peer-to-peer traffic or even intentionally interfere with it. This has made clients increasingly worried over some ISPs actions, from traffic shaping (protocol/packet prioritization) to traffic tampering.

San Francisco-based branch of the Electronic Frontier Foundation (EFF) a digital rights group have successfully verified that this type of efforts by Internet providers to disrupt some uses of their services and evidences seem to indicate that it is an increasing trend other as reports have reached the EFF and verified by an investigation by The Associated Press.

EFF Releases Report Interference with Internet Traffic on ComCast (http://www.eff.org/wp/packet-forgery-isps-report-comcast-affair), other information is available about this subject on the EFF site.

Metrics

Forced traffic shaping

Some ISPs are now using more sophisticated measures (e.g. pattern/timing analysis or categorizing ports based on side-channel data) to detect P2P traffic. This means that even

77 http://en.wikibooks.org/wiki/%20similar%20solution%20for%20Vista%20SP1
78 http://badvista.org
79 http://badvista.fsf.org/what-s-wrong-with-microsoft-windows-vista

encrypted traffic can be throttled. However, with ISPs that continue to use simpler, less costly methods to identify and throttle P2P networks, the current countermeasures added to P2P solutions remains extremely effective.

Traffic tampering

Traffic tampering is more worrying than Traffic shaping and harder to be noticed or verified. It can also be defined as spoofing[80], consisting in the injection of adulterated/fake information into communication by gaming a given protocol. It like the post office taking the identity of one of your friends and sending mail to you in it's name.

Pcapdiff (`http://www.eff.org/testyourisp/pcapdiff/`) is a free Python tool developed by the EFF to compare two packet captures and identify potentially forged, dropped, or mangled packets.

When there is a demand offers to satisfy it quickly fallow. One example is the Sandvine Incorporated[81] application that is able to intercept p2p communications and subvert the protocols. This type of application has dual proposes (for the owner perspective), for instance the Sandvine application was primarily designed to change Gnutella network traffic as a path "optimizer". But as today the adoption of the BitTorrent seems to be taking primacy, recent offers of the Sandvine application are capable of intercepting BitTorrent peer-to-tracker communication as to identify peers based on the IP address and port numbers in the peer list returned from the tracker. When Sandvine later sees connections to peers in the intercepted peer lists, it may (according to policy) break these connections by sending counterfeit TCP resets. Even if BitTorrent protocol continues to implement countermeasures, they have costs and it turns the problem into an "arms race", the issue is moral or legal in nature, with security implications.

The fight for network neutrality

Network Neutrality deals with the need to prevent ISPs from double dipping on charges/fees for both the clients paying for their broadband connections and WEB sites/Organizations having also to pay for prioritization of traffic according to origination and destination or protocol used.

1.5.3 The secretive Anti-Counterfeiting Trade Agreement

The **Anti-Counterfeiting Trade Agreement**[82] (**ACTA**) is a proposed plurilateral agreement[83] for the purpose of establishing international standards on intellectual property[84] rights enforcement.

80 `http://en.wikipedia.org/wiki/Spoofing%20attack`
81 `http://en.wikipedia.org/wiki/Sandvine%20Incorporated`
82 `http://en.wikipedia.org/wiki/Anti-Counterfeiting%20Trade%20Agreement`
83 `http://en.wikipedia.org/wiki/plurilateral%20agreement`
84 `http://en.wikipedia.org/wiki/intellectual%20property`

ACTA has the purpose to establish a new international legal framework that countries can join on a voluntary basis and would create its own governing body outside existing international institutions such as the World Trade Organization[85] (WTO), the World Intellectual Property Organization[86] (WIPO) or the United Nations[87].

The idea to create a plurilateral agreement on counterfeiting was developed by Japan[88] and the United States[89] in 2006. Canada, the European Union[90] and Switzerland[91] joined the preliminary talks throughout 2006 and 2007. Official negotiations began in June 2008, with Australia[92], Mexico[93], Morocco[94], New Zealand[95], the Republic of Korea[96] and Singapore[97] joining the talks. It is planned for negotiations to finish in September 2010.

Negotiating countries have described it as a response "to the increase in global trade of counterfeit goods and pirated copyright protected works.

The scope of ACTA is broad, including counterfeit[98] goods, generic[99] medicines and copyright infringement[100] on the Internet[101]. Because it is in effect a treaty, ACTA would overcome many court precedents defining consumer rights as to "fair use" and would either change or remove limitations on the application of intellectual property laws.

After a series of draft text leaks in 2008, 2009 and 2010 the negotiating parties published the official version of the current draft[102] on 20 April 2010.

United States

Both the Obama administration and the Bush administration had rejected requests to make the text of ACTA public, with the White House saying that disclosure would cause "damage to the national security."

In 2009, Knowledge Ecology International filed a FOIA (*Freedom of Information Act*) request in the United States, but their entire request was denied. The Office of the United States Trade Representative's Freedom of Information office stated the request was withheld for being material "properly classified in the interest of national security."

85 http://en.wikipedia.org/wiki/World%20Trade%20Organization
86 http://en.wikipedia.org/wiki/World%20Intellectual%20Property%20Organization
87 http://en.wikipedia.org/wiki/United%20Nations
88 http://en.wikipedia.org/wiki/Japan
89 http://en.wikipedia.org/wiki/United%20States
90 http://en.wikipedia.org/wiki/European%20Union
91 http://en.wikipedia.org/wiki/Switzerland
92 http://en.wikipedia.org/wiki/Australia
93 http://en.wikipedia.org/wiki/Mexico
94 http://en.wikipedia.org/wiki/Morocco
95 http://en.wikipedia.org/wiki/New%20Zealand
96 http://en.wikipedia.org/wiki/Republic%20of%20Korea
97 http://en.wikipedia.org/wiki/Singapore
98 http://en.wikipedia.org/wiki/counterfeit
99 http://en.wikipedia.org/wiki/generic%20drug
100 http://en.wikipedia.org/wiki/copyright%20infringement
101 http://en.wikipedia.org/wiki/Internet
102 http://trade.ec.europa.eu/doclib/docs/2010/april/tradoc_146029.pdf

US Senators Bernie Sanders (I-VT)[103] and Sherrod Brown (D-OH)[104] penned a letter on 23 November 2009, asking the United States Trade Representative to make the text of the ACTA public.

Secret negotiations

The Electronic Frontier Foundation[105] (EFF) opposes ACTA, calling for more public spotlight on the proposed treaty in its paper "Sunlight for ACTA" (`http://www.eff.org/action/sunlight-acta`)

Since May 2008 discussion papers and other documents relating to the negotiation of ACTA have been uploaded to Wikileaks, and newspaper reports about the secret negotiations swiftly followed.

In June 2008 Canadian academic Michael Geist writing for *Copyright News* argued that "Government Should Lift Veil on ACTA Secrecy" noting before documents leaked on the Internet ACTA was shrouded in secrecy. Coverage of the documents by the Toronto Star[106] "sparked widespread opposition as Canadians worry about the prospect of a trade deal that could lead to invasive searches of personal computers and increased surveillance of online activities." Geist argues that public disclosure of the draft ACTA treaty "might put an end to fears about iPod searching border guards" and that it "could focus attention on other key concerns including greater Internet service provider filtering of content, heightened liability for websites that link to allegedly infringing content, and diminished privacy for Internet users." Geist also argues that greater transparency would lead to a more inclusive process, highlighting that the ACTA negotiations have excluded both civil society groups as well as developing countries. Geist reports that "reports suggest that trade negotiators have been required to sign non-disclosure agreements for fear of word of the treaty's provisions leaking to the public." He argues that there is a need for "cooperation from all stakeholders to battle counterfeiting concerns" and that "an effective strategy requires broader participation and regular mechanisms for feedback".

In November 2008 the European Commission responded to these allegations as follows:

> It is alleged that the negotiations are undertaken under a veil of secrecy. This is not correct. For reasons of efficiency, it is only natural that intergovernmental negotiations dealing with issues that have an economic impact, do not take place in public and that negotiators are bound by a certain level of discretion. However, there has never been any intention to hide the fact that negotiations took place, or to conceal the ultimate objectives of the negotiations, the positions taken in European Commission Trade 5/6 the negotiations or even details on when and where these negotiations are taking place. The EU and other partners (US, Japan, Canada, etc.) announced their intention to start negotiations of ACTA on 23 October 2007, in well publicised press releases. Since then we have talked about ACTA on dozens of occasions, including at the European Parliament (INTA committee meetings), and in numerous well attended

103 `http://en.wikipedia.org/wiki/Bernie%20Sanders`
104 `http://en.wikipedia.org/wiki/Sherrod%20Brown`
105 `http://en.wikipedia.org/wiki/Electronic%20Frontier%20Foundation`
106 `http://en.wikipedia.org/wiki/Toronto%20Star`

seminars. Commission organised a stakeholders' consultation meeting on 23 June in Brussels, open to all – industry and citizens and attended by more than 100 participants. US, Australia, Canada, New Zealand and other ACTA partners did the same.

This position changed in 10 March 2010 with a direct European Parliament resolution criticizing the ACTA, the proceedings and the infringements on fundamental human rights.

Threats to freedom and fundamental human rights

An open letter signed by many organizations, including Consumers International[107], EDRi[108] (27 European civil rights and privacy NGOs), the Free Software Foundation[109] (FSF), the Electronic Frontier Foundation (EFF), ASIC (French trade association for web 2.0 companies), and the Free Knowledge Institute[110] (FKI), states that "the current draft of ACTA would profoundly restrict the fundamental rights and freedoms of European citizens, most notably the freedom of expression and communication privacy."

The Free Software Foundation[111] argues that ACTA will create a culture of surveillance and suspicion. (see `http://www.fsf.org/campaigns/acta` "Speak out against ACTA").

Aaron Shaw, Research Fellow at the Berkman Center for Internet & Society[112] at Harvard University[113], argues that "ACTA would create unduly harsh legal standards that do not reflect contemporary principles of democratic government, free market exchange, or civil liberties. Even though the precise terms of ACTA remain undecided, the negotiants' preliminary documents reveal many troubling aspects of the proposed agreement" such as removing "legal safeguards that protect Internet Service Providers from liability for the actions of their subscribers" in effect giving ISPs no option but to comply with privacy invasions. Shaw further says that ACTA would also facilitate privacy violations by trademark and copyright holders against private citizens suspected of infringement activities without any sort of legal due process".

The Free Software Foundation (FSF) has published "Speak out against ACTA", stating that the ACTA threatens free software by creating a culture "in which the freedom that is required to produce free software is seen as dangerous and threatening rather than creative, innovative, and exciting."

ACTA would also require that existing ISP no longer host free software that can access copyrighted media; this would substantially affect many sites that offer free software or host software projects such as SourceForge[114]. Specifically the FSF argues that ACTA will make it more difficult and expensive to distribute free software via file sharing[115] and

107 `http://en.wikipedia.org/wiki/Consumers%20International`
108 `http://en.wikipedia.org/wiki/EDRi`
109 `http://en.wikipedia.org/wiki/Free%20Software%20Foundation`
110 `http://en.wikipedia.org/wiki/Free%20Knowledge%20Institute`
111 `http://en.wikipedia.org/wiki/Free%20Software%20Foundation`
112 `http://en.wikipedia.org/wiki/Berkman%20Center%20for%20Internet%20%26%20Society`
113 `http://en.wikipedia.org/wiki/Harvard%20University`
114 `http://en.wikipedia.org/wiki/SourceForge`
115 `http://en.wikipedia.org/wiki/file%20sharing`

P2P[116] technologies like BitTorrent[117], which are currently used to distribute large amounts of free software. The FSF also argues that ACTA will make it harder for users of free operating systems to play non-free media because DRM[118] protected media would not be legally playable with free software.

On 10 March 2010, the European Parliament adopted a resolution criticizing the ACTA[119], with 663 in favor of the resolution and 13 against, arguing that "in order to respect fundamental rights, such as the right to freedom of expression and the right to privacy" certain changes in the ACTA content and the process should be made.

Legal scope

Nate Anderson with Ars Technica[120] pointed out in his article (http://arstechnica.com/news.ars/post/20080602-the-real-acta-threat-its-not-ipod-scanning-border-guards.html), that ACTA encourages service providers to provide information about suspected infringers, by giving them "safe harbor from certain legal threats". Similarly, it provides for criminalization of copyright infringement, granting law enforcement the powers to perform criminal investigation, arrests and pursue criminal citations or prosecution of suspects who may have infringed on copyright. It also allows criminal investigations and invasive searches to be performed against individuals for whom there is no probable cause, and in that regard weakens the presumption of innocence and allows what would in the past have been considered unlawful searches.

Since ACTA is an international treaty, it is an example of policy laundering[121] used to establish and implement legal changes. Policy laundering allows legal provisions—usually administered through public legislation and subject to judiciary oversight—to be pushed through via closed negotiations among members of the executive bodies of the signatories. Once ratified, companies belonging to non-members may be forced to follow the ACTA requirements since they will fall out of the safe harbor protections. Also, the use of trade incentives and the like to persuade other nations to adopt treaties is a standard approach in international relationships. Additional signatories would have to accept ACTA's terms without much scope for negotiation.

From 16–18 June 2010, a conference was held at the Washington College of Law[122], attended by "over 90 academics, practitioners and public interest organizations from six continents". Their conclusions were published on 23 June 2010 on the American University Washington College of Law website. They found "that the terms of the publicly released draft of ACTA threaten numerous public interests, including every concern specifically disclaimed by negotiators."

116 http://en.wikipedia.org/wiki/Peer-to-peer
117 http://en.wikipedia.org/wiki/BitTorrent%20%28protocol%29
118 http://en.wikipedia.org/wiki/Digital%20rights%20management
119 http://www.europarl.europa.eu/sides/getDoc.do?type=TA&reference=P7-TA-2010-0058&language=EN&ring=P7-RC-2010-0154
120 http://en.wikipedia.org/wiki/Ars%20Technica
121 http://en.wikipedia.org/wiki/policy%20laundering
122 http://en.wikipedia.org/wiki/Washington%20College%20of%20Law

Requests for disclosure

In September 2008 a number of interest groups urged parties to the ACTA negotiations to disclose the language of the evolving agreement. In an open letter the groups argued that: "Because the text of the treaty and relevant discussion documents remain secret, the public has no way of assessing whether and to what extent these and related concerns are merited." The interest groups included: the Consumers Union[123], the Electronic Frontier Foundation, Essential Action, IP Justice[124], Knowledge Ecology International[125], Public Knowledge[126], Global Trade Watch[127], the US Public Interest Research Group, IP Left (Korea), the Canadian Library Association[128], the Consumers Union of Japan, the National Consumer Council[129] (UK) and the Doctors without Borders[130]' Campaign for Essential Medicines.

123 http://en.wikipedia.org/wiki/Consumers%20Union
124 http://en.wikipedia.org/wiki/IP%20Justice
125 http://en.wikipedia.org/wiki/Knowledge%20Ecology%20International
126 http://en.wikipedia.org/wiki/Public%20Knowledge
127 http://en.wikipedia.org/wiki/Global%20Trade%20Watch
128 http://en.wikipedia.org/wiki/Canadian%20Library%20Association
129 http://en.wikipedia.org/wiki/National%20Consumer%20Council
130 http://en.wikipedia.org/wiki/Doctors%20without%20Borders

2 P2P Networks and Protocols

This chapter will try to provide an overview of what is Peer-to-Peer, it's historical evolution, technologies and uses.

2.1 P2P and the Internet: A "bit" of History

P2P is not a new technology, P2P is almost as old as the Internet it started with the email and the next generation were called "metacomputing" or classed as "middleware". The concept of it took the Internet by storm only because of a general decentralization of the P2P protocols, that not only gave power to the simple user but also made possible savings on information distribution resources, a very different approach from the old centralization concept.

This can be a problem for security or control of that shared information, or in other words a "democratization" of the information (the well known use of P2P for downloading copies of MP3s, programs, and even movies from file sharing networks), and due to it's decentralizing nature the traffic patterns, are hard to predict, so, providing infrastructures to support it is a major problem most ISPs are now aware.

P2P has also been heralded as the solution to index the deep Web[1] since most implantations of P2P technologies are based and oriented to wired networks running TCP/IP[2]. Some are even being transfered to wireless uses (sensors, phones and robotic applications), you probably have already heard of some military implementation of intelligent mines or robotic insect hordes.

Ultimately what made P2P popular was that it created a leveled play field, due to the easy access to computers and networking infrastructures we have today in most parts of the world. We are free to easily become producers, replacing the old centralized models where most of the population remained as consumers dependent on a single entity (monopoly, brand, visibility) for the distribution or creation of services or digital goods. This shift will undoubtedly reduce the costs of production and distribution in general as the price of services and products that can be digital transfered, the cost is now also becoming evident the quality will also be downgraded until a new system for classification emerges, this can be seen today in relation to the written media after the Internet impact.

1 http://en.wikipedia.org/wiki/deep%20Web

2 http://www.tcpipguide.com/free/t_TheTCPIPGuideIntroductionandGuideToTheGuide.htm

2.2 FIDO net

2.3 eMail

Electronic mail[3] (often abbreviated as e-mail or email), started as a centralized service for creating, transmitting, or storing primarily text-based human communications with digital communications systems with the first standardization efforts resulting in the adoption of the Simple Mail Transfer Protocol (SMTP), first published as Internet Standard 10 (RFC 821) in 1982.

Modern e-mail systems are based on a store-and-forward model in which e-mail computer server systems, accept, forward, or store messages on behalf of users, who only connect to the e-mail infrastructure with their personal computer or other network-enabled device for the duration of message transmission or retrieval to or from their designated server.

Originally, e-mail consisted only of text messages composed in the ASCII character set, today, virtually any media format can be sent today, including attachments of audio and video clips.

2.3.1 Peer2Mail

Peer to Mail (`http://www.peer2mail.com/`) is a FreeWare application for Windows that lets you store and share files on any web-mail account, you can use Web-mail providers such as Gmail (Google Mail), Walla!, Yahoo and others, it will split the shared files into segments that will be compressed and encrypted and then sends the file segments one by one to an account you have administration access. To Download the files the process is reversed.

Security

The ecryptation was broken in Peer2Mail v1.4 (prior versions are also affected) - Peer2Mail Encrypt PassDumper Exploit[4].

2.4 Usenet

Usenet is the original peer to peer file-sharing application. It was originally developed to make use of UUCP (Unix to Unix Copy) to synchronize two computers' message queues. Usenet stores each article in an individual file and each newsgroup in its own directory. Synchronizing two peers is as simple as synchronizing selected directories in two disparate filesystems.

Usenet was created with the assumption that everyone would receive, store and forward the same news. This assumption greatly simplified development to the point where a peer was

3 `http://en.wikipedia.org/wiki/E-mail`
4 `http://www.packetstormsecurity.org/0501-exploits/peer2mail.c`

able to connect to any other peer in order to get news. The fragmentation of Usenet into myriad newsgroups allowed it to scale while preserving its basic architecture. 'Every node stores all news' became 'every node stores all news in newsgroups it subscribes to'.

Of all other peer-to-peer protocols, Usenet is closest to Freenet since all nodes are absolutely equal and global maps of the network are not kept by any subset of nodes. Unlike Freenet, which works by recursive pulling of a requested object along a linear chain of peers, Usenet works by recursive pushing of all news to their immediate neighbors into a tree.

2.5 FTP

The File Transfer Protocol (FTP), can be seen as a primordial P2P protocol. Even if it depends on a client/server structure the limitation is only on the type of application (client/server) one run since the roles are flexible.

2.5.1 File eXchange Protocol (FXP)

2.6 Zero configuration networking

Zero configuration networking (zeroconf), is a set of techniques that automatically creates a usable Internet Protocol (IP) network P2P fashion, without manual operator intervention or special configuration servers.

2.6.1 Bonjour

Bonjour[5], formerly **Rendezvous**. A service discovery[6] protocol by Apple Inc.[7]'s. Bonjour locates in a P2P fashion, devices such as printers, as well as other computers and the services that those devices offer on a local network[8] using multicast[9] to maintain a Domain Name System[10] record. The software is built into Apple's Mac OS X[11] operating system[12] from version 10.2 onward, and can be installed onto computers using Microsoft Windows[13] operating systems. Bonjour also supports components that included other software, such as iTunes[14].

Bonjour for Windows (`http://support.apple.com/downloads/Bonjour_for_Windows`)

5 http://en.wikipedia.org/wiki/Bonjour%20%28software%29
6 http://en.wikipedia.org/wiki/service%20discovery
7 http://en.wikipedia.org/wiki/Apple%20Inc.
8 http://en.wikipedia.org/wiki/local%20area%20network
9 http://en.wikipedia.org/wiki/multicast
10 http://en.wikipedia.org/wiki/Domain%20Name%20System
11 http://en.wikibooks.org/wiki/Mac%20OS%20X
12 http://en.wikipedia.org/wiki/operating%20system
13 http://en.wikipedia.org/wiki/Microsoft%20Windows
14 http://en.wikipedia.org/wiki/iTunes

Bonjour for Windows includes a plugin to discover advertised HTTP servers using Internet Explorer. If you have Bonjour devices on your local network with embedded HTTP (Web) servers, they will appear in the list.

2.7 Internet Relay Chat (IRC)

Internet Relay Chat, commonly abbreviated **IRC**[15] is a real-time text-based multi-user communication protocol specification and implementation; it relays messages between users on the network. IRC was born sometime in 1988, from the mind of Jarkko Oikarinen[16]. According to IRChelp.org (`http://www.irchelp.org/irchelp/rfc/`), the official specification for IRC was written in 1993 in the RFC[17] format. The protocol was defined in the "RFC 1459: Internet Relay Chat Protocol", a really excellent source for both an introduction to and detailed information about the IRC protocol.

IRC's largest unit of architecture is the **IRC network**. There are perhaps hundreds of IRC networks in the world each one running parallel and disjoint from the others. A client logged into one network can communicate only with other clients on the same network, not with clients on other networks. Each network is composed of one or more **IRC servers**. An IRC client is a program that connects to a given IRC server in order to have the server relay communications to and from other clients on the same network but not necessarily the same server.

Messages on IRC are sent as blocks. That is, other IRC clients will not see one typing and editing as one does so. One creates a message block (often just a sentence) and transmits that block all at once, which is received by the server and based on the addressing, delivers it to the appropriate client or relays it to other servers so that it may be delivered or relayed again, et cetera. For a look into the messages exchanged on an IRC network you can take a look into (`http://www.alien.net.au/irc/irc2numerics.html),` it clearly identifies the several implementations and functions.

Once connected to a server, addressing of other clients is achieved through **IRC nicknames**. A nickname is simply a unique string of ASCII characters identifying a particular client. Although implementations vary, restrictions on nicknames usually dictate that they be composed only of characters a-z, A-Z, 0-9, underscore, and dash.

Another form of addressing on IRC, and arguably one of its defining features, is the **IRC channel**. IRC channels are often compared to CB Radio (Citizen's Band Radio) channels. While with CB one is said to be "listening" to a channel, in IRC one's client is said to be "joined" to the channel. Any communication sent to that channel is then "heard" or seen by the client. On the other hand, other clients on the same network or even on the same server, but not on the same channel will not see any messages sent to that channel.

Updated information on IRC can be obtained at IRC.org[18], the move to support IPv6 and the new technical papers, the IETF (Internet Engineering Task-Force) approved the most

15 `http://en.wikipedia.org/wiki/IRC`
16 `http://en.wikipedia.org/wiki/Jarkko%20Oikarinen`
17 `http://en.wikipedia.org/wiki/RFC`
18 `http://www.irc.org/`

current technical drafts (April 2000 - authored by C Kalt):

```
RFC 2810 : IRC Architecture
RFC 2811 : IRC Channel-Management
RFC 2812 : IRC Client-Protocol
RFC 2813 : IRC Server-Protocol
```

These documents are already available on IRC.org's official FTP-server, reachable at ftp://ftp.irc.org/irc/server

While IRC is by definition not a P2P[19] protocol, IRC does have some extensions that support text and file transmission directly from client to client without any relay at all. These extensions are known as DCC[20] (Direct Client to Client) and CTCP[21] (Client To Client Protocol).

Ident Protocol

The Ident Protocol[22], specified in RFC 1413[23], is an Internet protocol that helps identify the user of a particular TCP connection, and differentiate them from others sharing the same connection on the a server.

The Ident Protocol is designed to work it self as a server daemon[24], on a user[25]'s computer, where it receives requests to a specified port[26], generally 113. The server will then send a specially designed response that identifies the username of the current user.

Most standalone Windows machines do not have an Ident service running or present by default, in this case you may need to run your own Ident server (there are several stand alone servers available), on the other hand if you are on a Unix/Linux machine the service is there by default. Some Windows IRC clients have also an Ident server built into them.

The reason for having an running Ident server is due to IRC servers using the information for security reasons (not a particularly efficient way of doing), some going so far as blocking clients without an Ident response, the main reason being that it makes it much harder to connect via an "open proxy[27]" or a system where you have compromised a single account of some form but do not have root[28].

19 http://en.wikibooks.org/wiki/Internet_Technologies%2FPeer-to-Peer_%2528P2P%2529
20 http://en.wikipedia.org/wiki/Direct%20Client-to-Client
21 http://en.wikipedia.org/wiki/Client-to-client%20protocol
22 http://en.wikipedia.org/wiki/Ident
23 http://tools.ietf.org/html/rfc1413
24 http://en.wikipedia.org/wiki/daemon%20%28computer%20software%29
25 http://en.wikipedia.org/wiki/user%20%28computing%29
26 http://en.wikipedia.org/wiki/TCP%20and%20UDP%20port
27 http://en.wikipedia.org/wiki/open%20proxy
28 http://en.wikipedia.org/wiki/superuser

DCC (Direct Client Connect) Protocol

CTCP (Client To Client Protocol) Protocol

With CTCP, clients can implement commands such as "ctcp nickname version" or "ctcp nickname ping" to get some interesting infos about other users (like mIRC[29] does).

Bots or Robots

IRC systems also support (ro)bots, in this case they are not real users but a collection of commands that are loaded from a script (text) file into the IRC client, or even a stand alone program that connects to a IRC channel. They serve to ease the human interaction with the system, provide some kind of automation or even to test or implement some AI.

Basic Commands

Here are some basic commands for IRC:

Command	**What it does**	**Example**
/attach/server	Sign on to a server	/**attach** irc.freenode.net/**server** irc.freenode.net
/nick	Set your nickname	/**nick** YourName
/join	Join a channel	/**join** #wikibooks
/msg	Sends a message (can either be private or to the entire channel)	Message the channel: /**msg** #wikibooks hello world!Send a private message: /**msg** JohnDoe Hi john.
/whois	Display information about a user on the server	/**whois** JohnDoe
/clear/clearall	Clears a channel's text.Clears all open channel's text.	/**clear**/**clearall**
/away	Sets an away message. **Note: Type /away again to return from away.**	/**away** I'm away because...
/me	Sends an action to the channel. See example.	The following:/**me** loves pie.would output to the chat in the case of JohnDoe:**JohnDoe** loves pie.

Privileged User Commands

Commands for half-operators, channel operators, channel owners, and Admins:

29 http://en.wikipedia.org/wiki/mIRC

Command	What it does	Example
/kick	Kicks, or boots a user from the channel. **You must be a half-operator or greater to do this.**	Kick a user from the channel with a reason: /**kick** JohnDoe I kicked you because...
/ban/unban	Bans a user from the channel. **You must be a channel operator or greater to do this.**Unbans a user from the channel. **You must be a channel operator or greater to do this.**	/**ban** JohnDoe/**unban** JohnDoe

IRC Networks

- **EFnet**
- **Undernet**
- **Dalnet**
- **Rede Portuguesa de IRC (PTnet)** (`http://www.ptnet.org/`), is the biggest Portuguese IRC network, it was created in 1997, you can get an updated list of it's server (`http://www.ptnet.org/servidores`).

2.7.1 Security Risks

Software Implementations

- **KVIrc**[30] (`http://www.kvirc.net/`) an open source (GPL) portable IRC client based on the Qt GUI toolkit and coded in C++.
- **Bersirc**[31] (`http://bersirc.free2code.net/index.php/home/`), an open source IRC client (LGPL), coded in C, that runs on Windows (Linux and Mac OS X ports under development) by utilizing the Claro GUI Toolkit.
- **XChat**[32] (`http://www.xchat.org/`) is an IRC (chat) program for Windows and UNIX (Linux/BSD) operating systems. I.R.C. is Internet Relay Chat. XChat runs on most BSD and POSIX compliant operating systems. Open Source (GPL), coded in C.
- **Irssi**[33] (`http://irssi.org/`), an IRC client program originally written by Timo Sirainen, and released under the terms of the GNU General Public License. It is written in the C programming language and in normal operation uses a text-mode user interface.
- **mIRC**[34] (`http://www.mirc.co.uk/`), a shareware Internet Relay Chat client for Windows, created in 1995 and developed by Khaled Mardam-Bey. This was originally its only use, but it has evolved into a highly configurable tool that can be used for many purposes due to its integrated scripting language.

30 `http://en.wikipedia.org/wiki/KVIrc`
31 `http://en.wikipedia.org/wiki/Bersirc`
32 `http://en.wikipedia.org/wiki/X-Chat`
33 `http://en.wikipedia.org/wiki/Irssi`
34 `http://en.wikipedia.org/wiki/mIRC`

You can also check Wikipedia list of IRC clients[35] and Comparison of Internet Relay Chat clients[36] (not up-to-date)...

Invisible IRC Project

A technological advancement in relation to normal IRC networks, created by invisibleNET, a research & development driven organization whose main focus is the innovation of intelligent network technology. Its goal is to provide the highest standards in security and privacy on the widely used, yet notoriously insecure Internet.

Invisible IRC Project (`http://invisibleip.sourceforge.net/iip/`) is a three-tier, peer distributed network designed to be a secure and private transport medium for high speed, low volume, dynamic content. Features:

- Perfect Forward Security using Diffie-Hellman Key Exchange Protocol
- Constant session key rotation
- 128 bit Blowfish node-to-node encryption
- 160 bit Blowfish end-to-end encryption
- Chaffed traffic to thwart traffic analysis
- Secure dynamic routing using cryptographically signed namespaces for node identification
- Node level flood control
- Seamless use of standard IRC clients
- Gui interface
- Peer distributed topology for protecting the identity of users
- Completely modular in design, all protocols are plug-in capable

The IIP software is released under the GPL license and is available for Windows 98/ME/NT/2000/XP, *nix/BSD and Mac OSX, coded in C.

2.8 Instant Messaging

Instant messaging can be considered a subtype of P2P, in simple terms it consists on the act of instantly communicating between two or more people over a network (LAN or WAN) mostly using text. This requires the use of a client program so that when a message is sent, where a notification is shown a short time after on the destination application, enabling it's user to reply to the original messages. IM protocols can be centralized or distributed or a mix of the two.

Instant messaging allows users to send quick notes or reminders to other users in almost real time. IM can, but may not, include any P2P implementation or support extra P2P services like, file-sharing, VoIP or Video Conference the broad definition is that IM is the almost instantaneous trading of messages, whatever form it takes.

Since any P2P network depends on participation, supporting some kind of IM implementations is very important as a way to create a community and sustain the network.

35 `http://en.wikipedia.org/wiki/List%20of%20IRC%20clients`
36 `http://en.wikipedia.org/wiki/Comparison%20of%20Internet%20Relay%20Chat%20clients`

2.8.1 Security Risks

2.8.2 IM Software Implementations

- Gaim/Pidgin[37] (`http://pidgin.im/pidgin/home/`) OpenSource (GPL) instant messaging client supporting Windows, GNU, BSD, and many Unix derivatives and compatible with AIM, ICQ, MSN, Yahoo!, IRC, Jabber, Gadu-Gadu, SILC, GroupWise Messenger, and Zephyr networks.
- **Trillain** (`http://www.ceruleanstudios.com/`) skinnable chat client that supports AIM, ICQ, MSN, Yahoo!, and IRC, it also includes many features not included in those chat programs.
- **BitWise IM** (`http://www.bitwiseim.com),` Encrypted Cross-Platform (Windows, Mac OS X, and Linux) Instant Messaging, free but closed source, uses wxWidgets[38]. Also supports a Whiteboards, Voice Chat.
- **digsby** (`http://www.digsby.com),` a closed source, windows only, multiprotocol IM client that lets you chat with all your friends on AIM, MSN, Yahoo, ICQ, Google Talk, and Jabber with one simple to manage buddy list.
- **Google Talk** (`http://www.google.com/talk/),` Windows XP+ only, closed source, supports IM and interacts with the Gmail (google WEB mail) platform.

2.9 VoIP

Voice over IP can also be seen like an extension of a IM were text is substituted by live audio or video, the technological challenges are very similar, if not considering the type data of data that needs to be transfered and specific considerations due to timings. It is not uncommon for IM applications to also support VoIP or video conferencing.

Security on VoIP faces the same vulnerabilities and security threats of other P2P protocols and applications, including fuzzing, floods, spoofing, stealth attacks and VoIP spam.

2.10 Napster

The Napster network was created at application-level using a client-server protocol over point-to-point TCP. The server was in this case a centralized directory that would hold an index of all files offered (MP3/WMA). The clients would connect to the server, identify themselves to the server (users had an account on the server) and send a list of MP3/WMA files they were sharing to it enabling other clients to search that central repository for any file on the network and then request it from any available source.

Napster protocol specifications[39]

37 `http://en.wikipedia.org/wiki/Pidgin%20%28software%29`
38 `http://en.wikipedia.org/wiki/wxWidgets`
39 `http://opennap.sourceforge.net/napster.txt`

Software Implementation

- OpenNap[40] (`http://opennap.sourceforge.net`), a Napster based peer-to-peer, created as open source (GPL), in C using Win32, and so for Windows. Intended to extend the Napster protocol to allow sharing of any media type, and adding the ability to link servers together. Discontinued.
- **audioGnome** (`http://www.audiognome.com`), closed source but as freeware for Windows.
- **JNerve** (`http://jnerve.sourceforge.net`), an open source (GPL) Java Napster Server Protocol implementation with a goal of cross-platform compatibility.
- **Napsack** (`http://napsack.sourceforge.net`) is a specialized multi-threaded client for broadcasting Napster queries across multiple servers; the list of target servers is retrieved from www.napigator.com, and is user-filterable (based on the number of users, files, or gigs indexed). Opensourced (GPL) using Java.

2.11 Gnutella

Gnutella is an open file sharing Network[41] originally created by Justin Frankel of Nullsoft. That means, unlike most other networks, everyone can write a client which can access the GNet, if it fulfills the publicly available specifications.

The specifications are discussed and created by the GDF[42] (the gnutella development forum), an open Mailinglist for developers, with by this date over 1000 members. After that they are documented in the rfc-gnutella[43]. In this way all programs share a common base, while the protocol also allows for client specific options. The developers are careful to ensure the greatest possible backwards compatibility.

Despite the name, Gnutella isn't GNU-Software, though some Gnutella clients are GPL-licensed. It is an open network, and the origin of its name may be found more easily by eating too much Nutella, than at GNU. (That means: Gnutella is not a project of the FSF or related to GNU software tools). And while Gnutella was initially stated as a fully-distributed information-sharing technology, later versions of the protocol are a mix of centralized and distributed networks with "Servers" (Ultra or Super peers) and "Clients" (Leafs or Nodes).

A Gnutella client software is basically a mini search engine (offering an alternative to web search engines) and file serving system in one. Gnutella in new implementations also supports Tiger tree hashing (TTH) for for file transfers.

One sibling of Gnutella deserves special attention, even though some developers of current clients would deny it. It is called MP (Mikes Protocol) or the Shareaza Protocol by most Gnutella developers, while its developer called it Gnutella2, a name which gave his program (Shareaza) much media coverage and created and creates much controversy and aversion against it in the_gdf.

40 `http://en.wikipedia.org/wiki/Opennap`
41 `http://en.wikipedia.org/wiki/Filesharing`
42 `http://groups.yahoo.com/group/the_gdf`
43 `http://rfc-gnutella.sf.net`

Gnutella2 (Mike's Protocol,G2)

The result of a fork[44] of the Gnutella protocol, due to the failing of the developers community to reach a consensus on the evolution of the protocol.

Gnutella2[45] is also called Mike's Protocol since the first changes and implementation resulted from a single developer Michael Stokes[46]. In November 2002, Michael Stokes[47] formally and unilaterally announced the creation Gnutella2 protocol to the Gnutella Developers Forum[48], which caused a schism among the developers, and lead for the modifications no to be supported in several Gnutella applications since the original proposal did conflict with other vendors concepts (in specific LimeWire and Bearshare).

The now resulting implementation drops all of the old Gnutella protocol except for the connection handshake and adopts an entirely new search algorithm[49]. Gnutella2 is often abbreviated as **G2**.

2.11.1 Network model

The Original: FoF

You can imagine the original model of the Gnutella network as friends phoning each other to get information. One asks five others, each of whom asks 5 others and so on. After the first step the number of people reached is 5, after the second it is 25, after the 5th 3125, after the 7th 78,125 and after the 14th about 6.1 billion. That would be enough to reach every human being on this planet. The original Gnutella used 7 steps (called HTL: Hops To Live).

The biggest problem with this model (among others) is that you have to be a part of the clique before you can use it.

Problems of the FoF-model

The Friend-of-a-Friend model has certain disadvantages, which have their source in the way searches are performed. If a search brings too many results, the nodes, through which you are connected (your nearest 5 friends) can get overloaded, because every answer has to go through them, for they don't give out your \"phone number\", but their own and hand the answer to you. If you ask for the name of the head of University in the campus, you'll get hundreds of answers in reality, and thousands to millions on the web. Also, if every question is passed to every one in a 75,000 to 600,000 computer-network, and every computer asks only once an hour, each of them has to answer about 130 to 1600 questions per second. And they have to pass them on. While computers are fast, and today's Internet connections can handle quite a lot when compared to the connections a few years ago, this

44 http://en.wikipedia.org/wiki/Fork%20%28software%20development%29
45 http://en.wikipedia.org/wiki/Gnutella2
46 http://en.wikipedia.org/wiki/Michael%20Stokes
47 http://en.wikipedia.org/wiki/Michael%20Stokes
48 http://en.wikipedia.org/wiki/Gnutella%20Developers%20Forum
49 http://en.wikipedia.org/wiki/algorithm

is too much even for them. Just imagine your phone ringing endlessly the whole day for all kind of questions.

Getting in

To address the connection problem several ideas to solve have come about.

The first way: Pong-Caching
Pong Caching means that the node (aka you) asks its friends who their friends are. It means your friends introduce you to their friends, especially friends whom they value highly, and you write all new addresses in your phone-book, so you know whom to phone when your original friends are on holiday (Somehow like being at a continuous cocktail party). It is easy and has the advantage of giving you very reliable contacts, but there is no way of getting into the network without knowing at least one contact who is already in the net. That means you can always get back in, but won't be able to connect if you never did before.

The second way: Remember who answers
The second way is really simple. When one of your 5 friends calls back to say Smith (whom you didn't know before) knows something, you note her number. When you call him the next time as one of your five direct contacts, the chance is greater that you will get your information more quickly, because he will likely have friends who have similar interests to you (where else should he have gotten the information?), and those are more likely to have your information than randomly picked persons (at least when you ask about something similar to your last question). The drawback is that those contacts might not be at home often, so it might be that you find a contact with great knowledge, but whom you'll never be able to reach again. Still no way of getting in the first time. And now we get to one of the recent developments in Gnutella: GWebCaches. I will discuss them in the next part.

The third way: GWebCaches
To stay within the picture, a GWebCache is a contact who puts her phone number into the newspapers and keeps a record of those who call. When you've been away for some time and are no longer certain if your contacts still have the same mobile-phone numbers, you call the publicly known contact. Before giving you numbers, he/she will ask you: "Do you know other publicly known contacts? If yes, please tell me their numbers." That is done because they can't read all the newspapers, and you do it all the time without working too hard for it. That way, they keep track of each other. Then the contact gives you some numbers to call and notes your number (to give it to someone else) and the addresses of other public contacts he/she knows (GWebCaches).

This is roughly the way GWebCaches work. GwebCaches are essential only for the first connections. You use them when your local address-book is empty. They must not be preferred over your local address-book. As I stated, they are one of the new developments in Gnutella, and thus I will now get to some more of the recent changes within Gnutella and to future plans.

Ultrapeers and Leafs

Change who calls whom

You'll surely have friends who know very many other people, and whom you can ask, and be sure they'll know exactly the person who can give you the answer. These are called Ultrapeers in Gnutella. An Ultrapeer doesn't have to know much himself/herself, he/she just needs to know who knows it. In Gnutella that means that a good Ultrapeer doesn't need to have many files to benefit the network. If you're afraid to share much, you should become an Ultrapeer in Gnutella

In more detail

In the Computer World, as in the Real World, there are contacts who can cope with more calls, and those who can't phone often (or can't afford the bills). In the Real World this is because they have more free time. Whereas, in the Computer World, it is because they have faster connections (Like DSL, Cable, T1, T3 or similar broadband). Upon realizing this, the developers decided to change the topology, that means how the network looks from the outside when you draw it. Now you don't just call any of your friends, but only those whom you know have the time to take your call and to send it on to others. To save you from too many calls, they then ask you which kinds of informations you have or, to express it in a more human way, what your specialty is. In the Computer World that means your computer sends a list of all its files to the Ultrapeer, which is how we call these kinds of contacts. That list contains summaries (Hash-Strings) of all your shared files (those you decide to let others download) by which the downloader can verify that they are, indeed, the files he or she wants. Whenever a call reaches the Ultrapeer, it checks if you could know an answer and calls you only in that case.

These Ultrapeers have many connections to others, which means they have a really big address book. Normally they stay in contact with 16 other Ultrapeers whom they have in their address book and to whom they send questions, and who send them to 16 more, each. Also they have about 16 leafs, who can't or don't want to phone that much, from which they accept calls, and whose files or, for the human world, specialties they know.

This may seem like a foul bargain for the Ultrapeers, who devote far more resources to keeping the network intact than leafs, but in fact it isn't. While the Ultrapeer (UP) uses much of her time for keeping the network running, the leafs specialize on gathering and delivering information. So, when anyone, Ultrapeer or Leaf, wants to know something, he or she simply starts a call and a leaf specialist can explain it to them. That way people specialize to get more for all.

Intra-ultrapeer QRP

While with Ultrapeers not everyone needs to participate in sending questions to others, and people can specialize in sharing their information instead, the Ultrapeers would still send every question to everyone, without ever taking into account if that UP even has leaves, who have the files. This sounds normal, for how can an Ultrapeer know which files the other Ultrapeers have? The answer comes, again, from real life. A normal person knows her friends, and knows who of them might know the answer to a specific question, and who most surely will not. In Real Life this is mostly done through friendly chatting.

Now, computers normally don't chat idly, so they don't exchange this information by the way. Thus the Query Routing Protocol was developed. There each Leaf tells its Ultrapeers which files it has, but instead of taking the names, which would consume too much space, each word which is part of the name of a file is saved as numbers (these are computers after all). You can imagine this process like a game of BattleShips[50] (the numbers form the board with two coordinates). An Ultrapeer doesn't send all questions to a leaf, but only those which it might be able to answer (which hit a ship), and so Leafs get far less needless calls.

Now when this takes so much pressure off the leaves, why not extend it? Exactly that was done. Now all Ultrapeers send their boards to their direct neighbors. They send only those searches, which have one more step to go, to other Ultrapeers on whose board they score a hit. That means, the last two steps of a search will only be taken when there is a chance that they give results. You can see quite simply why this heavily reduces the bandwidth usage: imagine a tree, a normal tree, not one of those mathematical constructs. If you try to count the leaves, you have almost no chance. But if you take the leaves away and count only the branches, you have far less work to do. If you now take away all those tiny branches, you can really begin to count them. QRP doesn't take all leaves and all tiny branches away, but it removes those of them who couldn't give you an answer. Since every part through which a question has to travel consumes bandwidth, and there are far more leaves than branches, taking away, in many cases, many of the last two steps (that means many of the leaves and the tiny branches) reduces the number of questions the computers have to send on (there are far more leaves, than branches). The example doesn't work for all of Gnutella, but here it fits nicely.

Searching: Dynamic Querying

Now, while the Ultrapeer model and QRP partly solve the problem that you don't have the time to explain something properly to someone else, or to get it explained, because the phone rings endlessly for questions to which you know no answer (or in Tech-Speech: because the network-traffic exceeds your connection-speed), there is still another problem which might normally not even be visible, if you look at it. In the Real World, an Ultrapeer will ask for a specialist who can give you the info until he/she finds one, and then stops. In the Computer-World, the question is sent on and on, to as many contacts as possible, without looking if there already are answers.

With dynamic Querying that changes. Now the Ultrapeers ask one other Ultrapeer at a time, and wait a bit, to see if they get answers. When they have enough answers to be satisfied, they stop asking for more. It sounds pretty natural, but was quite a big step for Gnutella because it saves resources which were wasted on very popular questions. I'll take the example of the of the head of university again: now, if you ask for the head of university, your Ultrapeers will first see if they know someone directly who can answer your question. Then they will simply give you some numbers of people they know who live on the campus. You will still get more than one answer because they will give you more than one number, as they can't be sure that you'll reach every number they gave you. But you won't get thousands of phone-numbers (one from every student on the campus), first because the Ultrapeers would waste their time with that on something which doesn't give you additional

50 `http://en.wikipedia.org/wiki/Battleship%20game`

benefit, second, because you couldn't ever call all those people, and third, because then you might not reach your Ultrapeers anymore, because they would be so busy getting return calls from others who tell them numbers, and sending your question to other Ultrapeers.

Finding sources without searching aka the Download-Mesh

Now you might say, "but I can't download from those three, because others already do. I want to get all addresses from which I can download," (and you are not alone with this. I feel the same). By looking at the Real World, we can find a solution to this problem too, without having to waste too many resources on it. There (in the Real World), if you ask a specialist to explain something to you, and that specialist is busy, he or she will know some other specialists (because they know each other) who might have more time at the moment.

Getting this into Gnutella is not as easy as the Ultrapeer-Leaf Model nor as the Dynamic-Query Model. But the programmers found a way. As I stated at the Dynamic-Query-Model, you will get more than one number where you can ask. Now, when you call someone who should know the answer, you also give him or her the other numbers you know about. That way, the specialists will get to know each other (the same way, as the GWebCaches, which I mentioned before, learn of others of their kind). As everyone who asks also brings her own set of numbers, the specialists know more and more additional addresses, and when you ask them to explain, and they don't have time at the moment, they give them to you (they do it even if they have time, just in case they could be interrupted, and because in Gnutella you can download from more than one source at once, just like you can in the overnet-network (which does this to the extreme, but is only really efficient for big files). Additionally, the specialists also add you to their number of alternate-contacts, as soon as you know enough to teach others.

This is why often many people download files from you which you just downloaded yourself.

Downloading

Swarming and Partial File Sharing

Swarming is quickly explained (but hard to do in the friend-of-a-friend model, so I drop it for this part only). It works by simply getting one file from more than one person at once. The file is simply separated into several parts, as if you'd want to get a book from some friends and every one of them copied only a few pages of it. When you ask every one of them to copy a different part of the book, you'll get the complete book, and every one of them has only very little work to do (and if one doesn't have the time to do it, another one can).

Swarming works best with the Download Mesh and Partial File Sharing (PFS), which allows people to share files which they are downloading at the moment, because they can share those parts which they already have, while they still download from others. You can copy those pages which you have without having to have the whole book, since your pages are all numbered and friends can ask you for certain page numbers.

Downloading through Firewalls

Imagine there were people who couldn't be called, but could only call others (maybe because they only use public phones, or their number isn't displayed on your phone, and they don't like to give it out because they don't like being called by telemarketers, or by people terrorizing them over the phone). In Gnutella these are computers who are behind a firewall. They can call others and get information from them, but no one can call them.

The solution is to have the firewalled people call their Ultrapeers regularly and, when someone wants to call them, he or she simply calls the Ultrapeer who then holds two phones together, one to which the firewalled person (the one who can't be called) raised a call, and the one you called. That way you can talk to the firewalled person, but it takes two simultaneously running calls, which means, that it needs twice the bandwidth in the Computer-World. Firewalled persons always keep their connection to the Ultrapeers, who simply relay the information or data.

There are plans to save the Ultrapeers from this additional bandwidth usage by letting other people do the phone connecting. Then, when someone wanted to get information from a firewalled specialist, the Ultrapeer would tell the firewalled person and the asker to call a third person. That person would then hold the two phones together. In Gnutella most People have three to five phones, so this wouldn't be such a great problem. These phone-connectors will most likely be called routing-peers.

File-Magnets

File-Magnets stray from the Friend of a Friend model. They are links on Webpages, which you can simply click, and which will tell your file-sharing program to search Gnutella (in fact also other networks) for a specific file, and to download exactly this.

You can imagine it like an article in a newspaper which tells you information which gives your Ultrapeers the exact information that the specialist, from whom you want to learn, has to know. In the Real World you would most likely find one specialist and those who learned from him or her.

With a magnet-link you can avoid getting bad files because they use a hash-string, which is something like a summary of the information the specialist would give you. If he or she begins to tell you crap, you will see at once that it doesn't fit the summary. In Gnutella, the program asks for files to which the people who have them have assigned the same summary, aka Hash-string. After downloading, the program does its own summary and checks if they really match. If not, it tells you that the file is corrupt. The summaries from same files are always exactly the same because they are done via specific mathematic methods which always get to the same result when given the same data (aka information).

Different from Magnet-Links, KaZaA-Links and eDonkey-Links are not secure, because they use methods which can be betrayed with false files (for example a KaZaA-Link asks for a kind of summary which only checks the introduction and the first part of the information, but all the rest is ignored to make the summary quicker to create. Naturally it is very easy to give you false information, because they only have to tell the truth at the beginning,

then they can lie or fantasize as much as they want). Further information on Magnet-Links can be found here: Magnet-Uri[51] and on MagnetLink.org[52]

There is now a new version of magnet-links: KaZaA magnets. Sadly those aren't secure, for they use the KaZaA hashing system (the incomplete summary) with some changes (they now add another smaller summary, which might tell you about the missing parts, but they didn't publish, how they create it). If KaZaA-Magnets provide information about a search term, they might work with Gnutella, but they won't ensure that you get what they offer to you. If you find the word \"kzhash\"in the link, it might not be secure (aside from having a somewhat misplaced name).

2.11.2 Lime Wire LLC

LimeWire[53] a peer-to-peer file sharing[54] client for the cross-platform Java Platform[55], Open Source (GPL[56]), which uses the Gnutella[57] network to locate and transfer files. It also encourages the user to pay a fee, which will then give the user access to LimeWire Pro. Support for BitTorrent protocol is also present by using the C++, Boost licensed, libtorrent library.

2.11.3 Software Implementations

To become part of the Gnutella Network, you can use one of the clients listed below :

- **Deepnet Explorer** (`http://www.deepnetexplorer.com/`) a browser with RSS news reader, P2P client integration (Gnutella[58]) and phishing alarm, closed source, Windows only, freeware.
- **Phex**[59] cross-platform Java client.
- **XoloX**
- **Gnucleus**[60] - Gnutella, Gnutella2 (G2) coded using C++ and Microsoft MFC libraries. The Core is LGPL and communicates to a GPL frontend using Windows COM-base.
- **Gtk-Gnutella**[61], GPL, for GNU/Linux.
- **Hydranode** (multi-protocol, referenced on the eDonkey2000/eMule section)
- ezpeer[62], a Chinese client.
- pp365[63], a Chinese client.
- POCO[64], a Chinese, using GnucDNA.

51 `http://magnet-uri.sourceforge.net`
52 `http://www.magnetlink.org`
53 `http://en.wikipedia.org/wiki/LimeWire`
54 `http://en.wikipedia.org/wiki/peer-to-peer%20file%20sharing`
55 `http://en.wikipedia.org/wiki/Java%20Platform`
56 `http://en.wikipedia.org/wiki/GNU%20General%20Public%20License`
57 `http://en.wikipedia.org/wiki/Gnutella`
58 `http://en.wikipedia.org/wiki/Gnutella`
59 `http://www.phex.org/mambo/`
60 `http://www.gnucleus.com`
61 `http://gtk-gnutella.sourceforge.net`
62 `http://www.ezpeer.com.cn`
63 `http://www.pp365.com`
64 `http://www.poco.cn`

- Bearshare[65], free closed source for Window.
- CocoGnut[66], for RISC OS.
- Swapper[67], free closed source for Windows utilizing .NET.
- TrustyFiles[68], for Windows, supports FatTrack (KaZaA), Gnutella and G2.
- **Shareaza**[69] (`http://shareaza.sourceforge.net/`), Open Source (GPL[70]), coded in C++, MFC and ATL. Multi-network peer-to-peer file-sharing client supporting Gnutella2 (G2), Gnutella[71], eDonkey2000/eMule, BitTorrent, FTP and HTTP protocols.
- **FrostWire** (`http://sourceforge.net/projects/frostwire/`), a Peer to Peer (P2P) information sharing client for the Gnutella network. This project is not affiliated with LimeWire LLC. It is a fork of Limewire's Java implementation committed to never including content filters. FrostWires' source code (Java) is Licensed under the GNU GPL Open Source license. Newer version have moved to use the BitTorrent protocol.
  - Acquisition[72], a Mac OS X client based on the Limewire core, written in Cocoa, shareware.
  - XNap[73], a multi-network program in Java using the Limewire Core for Gnutella.

[74]

2.12 Ares Network

Ares (software implementation) was developed in the middle of 2002, originally using the Gnutella[75] network. After Six months of operation, it switched to its own network comprising the leaves-and-supernode[76]s p2p architecture. Having a protocol that can be difficult to identify made Ares at times the only P2P client that could functions on restricted networks, such as some university campuses.

2.12.1 Software Implementations

- **Ares**[77] (`http://aresgalaxy.sourceforge.net/`), a Chat/File Sharing P2P implementation in Delphi/Kylix. It's based on a Network organized into leafs and supernodes into a topology featuring broadcast-type searches. Ares can deliver a broader search horizon by means of the DHT technology, using a mime filter to DHT engine. Ares users can also join chat rooms or host a channel. It is for 32-bit MS Windows Operating Systems (NT/2000/XP) and Open Source under the GPL License (GNU General Public License).

65 `http://bearshare.com`
66 `http://www.alpha-programming.co.uk/software/cocognut/`
67 `https://www.revolutionarystuff.com/swapper/`
68 `http://www.trustyfiles.com`
69 `http://en.wikipedia.org/wiki/Shareaza`
70 `http://en.wikipedia.org/wiki/GNU%20General%20Public%20License`
71 `http://en.wikipedia.org/wiki/Gnutella`
72 `http://www.acquisitionx.com`
73 `http://xnap.sourceforge.net/`
74 `http://en.wikibooks.org/wiki/Category%3AThe%20World%20of%20Peer-to-Peer%20%28P2P%29`
75 `http://en.wikipedia.org/wiki/Gnutella`
76 `http://en.wikipedia.org/wiki/supernode%20%28networking%29`
77 `http://en.wikipedia.org/wiki/Ares%20Galaxy`

From version 1.9.0, data sharing was enabled between two peers behind a firewall. From version 1.9.4, Ares included support for the BitTorrent[78] protocol. From version 1.9.9, Ares Galaxy has support for the SHOUTcast[79] internet radio stations.

Discontinued Implementations

- *Warez P2P*[80] was a proprietary P2P[81] filesharing[82] service that uses the Ares network[83], and offers a service similar to that of Kazaa[84]. Up to version 1.6, Warez P2P was a clone of Ares Galaxy[85], created by Italian developer Alberto Trevisan, but since then has been developed independently by Neoteric Ltd until recently when it was discontinued.

86

2.13 Direct Connect

Direct connect is a peer-to-peer file sharing protocol/network but it uses a central server, this reliance on a central point can also be seen on the old Napster network, in that each server build an independent network (not an hybrid like for instance with eMule). One should note that some clients are now also implementing DHTs that will result in unifying used networks. The Direct Connect protocol was originally developed by Jonathan Hess for use on the Neo-Modus Direct Connect (NMDC) v1, released September 2001 and partially in NMDC v2, released in July 2003.

Direct Connect defines the servers as HUBs. Clients connect to a central hub and that hub feature a list of clients or users connected to it. Users can then search for files to download, or as chat with other users present (on that server).

Direct Connect also implements Tiger tree hashing (TTH) for for file transfers.

2.13.1 NMDC Protocol

created by Jon Hess at Neo-modus protocol mirror (`http://www.teamfair.info/wiki/index.php`)

2.13.2 ADC Protocol

The ADC protocol (`http://dcplusplus.sourceforge.net/ADC.html`) is similar to the Neo-Modus Direct Connect (NMDC) protocol. It consists of a text protocol for a client-server network, created with goal to be simple, yet extensible.

78 http://en.wikipedia.org/wiki/BitTorrent
79 http://en.wikipedia.org/wiki/SHOUTcast
80 http://en.wikipedia.org/wiki/Warez%20P2P
81 http://en.wikipedia.org/wiki/Peer-to-peer
82 http://en.wikipedia.org/wiki/filesharing
83 http://en.wikipedia.org/wiki/Ares%20Galaxy%23The%20Ares%20network
84 http://en.wikipedia.org/wiki/Kazaa
85 http://en.wikipedia.org/wiki/Ares%20Galaxy
86 http://en.wikibooks.org/wiki/Category%3AThe%20World%20of%20Peer-to-Peer%20%28P2P%29

Jon Hess contributed to the creation of this protocol with the original Direct Connect idea through the Neo-Modus Direct Connect client / hub. Other major contributing source was Jan Vidar Krey's DCTNG draft that lead to subsequent work by Dustin Brody, Walter Doekes, Timmo Stange, Fredrik Ullner, Fredrik Stenberg and others.

2.13.3 HUB Software Implementations

- **DConnect Daemon** (`http://www.dc.ds.pg.gda.pl/`), an open source Direct Connect's hub (working as daemon) written in C. Works currently under GNU Linux and FreeBSD, but is planned to be able to work on all Unixes and Windows. As a daemon it works in background and does not require any Xwindow system. Supports a telnet administration console.

2.13.4 Client Software Implementations

- **NeoModus Direct Connect**
- **DC++**[87] (`http://dcpp.net/` or `http://sourceforge.net/projects/dcplusplus/`), a project aimed at producing a file sharing client using the ADC protocol. It also supports connecting to the Direct Connect network, open source under GPL, using C++/MFC running on Windows. It is developed primarily by Jacek Sieka, nicknamed arnetheduck.
- **BCDC++**
- **RevConnect** (`http://www.revconnect.com/`), supports also Kademlia, Open Source under GPL, using C++/MFC running on Windows.
- **Strong DC++** (`http://strongdc.berlios.de/download.php?lang=eng`), Open Source under GPL, using C++/MFC running on Windows.
- **ApexDC++** (`http://www.peerwebdc.tk/ - http://sourceforge.net/projects/apexdc/`), based on Strong DC++. It has numerous features such as a plugin to block IP addresses (uses PeerGuardian blocklist), super seeding, customization, themes, and a nice Vista-type general user, Open Source under GPL, using C++/MFC running on Windows.
- **TkDC++** (`http://tkdcpp.com`), Open Source under GPL, using C++/MFC running on Windows.
- **GtkDC** (`http://gna.org/projects/gtkdc/),` a Direct Connect client written in C, based on a GIMP toolkit 2.0 GUI. It is intended to run on UNIX-based platforms, but with a little port it should be executable on Windows platform with Gtk libraries. Licensed under the GNU General Public License V2 or later.

2.14 eDonkey

eDonkey the original client for the eDonkey network[88] (also known as eDonkey2000 network or eD2k), was created and managed by MetaMachines (Sam Yagan and Jed McCaleh) based on the city of new York. It had a stable P2P community and the protocol was older than

87 `http://en.wikipedia.org/wiki/DC%20plus%20plus`
88 `http://en.wikipedia.org/wiki/eDonkey%20network`

BitTorrent it was created in 2002 shortly after the closing of Napster and competed with the FastTrack network. In June of 2005, the entertainment industry gained a victory in the Supreme Court (USA) that stated that every file-sharing developer could be sued for copyright infringement if they induced such behavior. In September 2005 the Recording Industry Association of America (RIAA) sent several commercial P2P developer cease and desist letters including to MetaMachines and with no founds to battle the interpretation of the Supreme Court decision, Sam Yagan conceded defeat as he testified to the United States Senate Judiciary Committee.

"...I am not here as an active participant in the future of P2P, but rather as one who has thrown in his towel and with no interest in replaying past issues..."
--Sam Yagan

On September 11, 2006 users could not get the eDonkey2000 client software, in September 12, 2006 MetaMachines settles for $30 Million (US) and the agreement closes any avenue MetaMachines had in dealing with any P2P technology in the future...

The eDonkey networks is centralized (as it depends on serves) to provide decentralized sharing of content (not stored on the servers), there are still many software implementations that support the network the most popular is eMule.

2.14.1 Protocol

"The eMule Protocol Specification" (`http://sourceforge.net/project/showfiles.php?group_id=53489&package_id=145950`) by Yoram Kulbak and Danny Bickson DANSS (Distributed Algorithms, Networking and Secure Systems) Lab - School of Computer Science and Engineering - The Hebrew University of Jerusalem, Israel - January 20, 2005, PDF document provided by the Emule Project.

2.14.2 Kademlia

Started as the Overnet[89] project by Jed McCaleb, the creator of eDonkey2000[90] to overcome the need of servers. Overnet implemented the Kademlia[91] algorithm. In late 2006, Overnet and all Overnet-owned resources were taken down as a result of legal actions from the RIAA[92] and others. However, since the core of Overnet is decentralized, Overnet clients are still able to function with limited functionality.

The KadC library (`http://kadc.sourceforge.net/`) provides an OpenSource C library to publishing and retrieving records in Kademlia-based Distributed Hash Tables.

89 `http://en.wikipedia.org/wiki/Overnet`
90 `http://en.wikipedia.org/wiki/eDonkey2000`
91 `http://en.wikipedia.org/wiki/Kademlia`
92 `http://en.wikipedia.org/wiki/RIAA`

A some what old paper named Kademlia: A Peer-to-peer Information System Based on the XOR Metric[93] by Petar Maymounkov and David Mazières can also be a source of information about the protocol.

The network is now known as Kademlia and is supported by many of the implementations of the old eDonkey/Overnet clients, especially by the eMule[94] project. Kademlia is a research effort to implement a full-featured peer-to-peer system based on the XOR[95] metric routing[96]. Of special interest are the objectives for efficient data storage and query; anonymity; network, content and user security and authentication.

2.14.3 eMule content database

(`http://content.emule-project.net/`) a service provided by the eMule project team for the eDonkey2000 and Kad network users, to make free content available for download and easy to find. The content database has been on line since around new years 2004.

2.14.4 Software Implementations

- **eMule** (`http://www.emule-project.net/`) a filesharing software implementation based on the eDonkey2000 network but offers more features than the standard client, open source C++/MFC and windows only, licensed under GPL (`http://sourceforge.net/projects/emule/`)
- **Xmod** (`http://savannah.nongnu.org/projects/x-mod/`) The Xmod is a Project is based on the eMule Client, OpenSource under the GPL.
- **xMule** (`http://www.xmule.ws/`), the X11 Mule, intended to bring a clone of eMule to virtually all the major Unix platforms, with a particular emphasis on Linux. C++ using wxWidgets for the GUI released as OpenSource under the GPL.
- **MLdonkey** (`http://mldonkey.sourceforge.net`) is a multi-platform, multi-network P2P implementation. It supports several large networks such as eDonkey, Overnet, Kademlia, Bittorrent, Gnutella (Bearshare, Limewire, etc.), Gnutella2 (Shareaza), or Fasttrack (Kazaa, Imesh, Grobster). Networks can be enabled or disabled. Searches are performed in parallel on all enabled networks. For some networks, each file can be downloaded from multiple clients concurrently.
- **AMule** (`http://www.amule.org/wiki/`) this project is based on the eMule Client, using also C++ but also wxWidgets and crypto++. Opensource under the GPL, currently

93 `http://www.cs.rice.edu/Conferences/IPTPS02/109.pdf`
94 `http://en.wikipedia.org/wiki/eMule`
95 `http://en.wikipedia.org/wiki/Exclusive%20or`
96 `http://en.wikipedia.org/wiki/Routing`

supports Linux, FreeBSD, OpenBSD, Windows, MacOS X and X-Box on both 32 and 64 bit computers.
- **eMule Bowlfish** (`http://pwp.netcabo.pt/DeepSea/`), another eMule based project that aims to provide an restricted Network solution.
- **Hydranode** (`http://hydranode.com/`) a modular, plugin-driven peer-to-peer client framework which is designed with true multi-network downloads in mind (Support for eDonkey2000 and Bittorrent networks). OpenSource under the GPL, supports Linux and Windows.
- **Shareaza** (multi-protocol, referenced on the Gnutella section)

2.15 BitTorrent

BitTorrent[97] is a protocol (BitTorrent Protocol Specification v1.0[98]) created by Bram Cohen[99] derived from the Gnutella concept, but primarily designed to distribute large computer files over the Internet and permit WEB integration, in fact it aimed to be a substitute for the old centralized HTTP downloads, not a full P2P network. As such it initially avoided the limitations of transmitting search request across the network, something that recently has been implemented with the adoption of a DHT similar to the eDonkey solution, that permits searches across a network. Making BitTorrent a two tiers P2P.

BitTorrent is used to distribute legitimate content but in itself does not make any differentiation about the copyright status of the shared material, as any other decentralized network this permits infringement on a massive scale. Bram Cohen and Ashwin Navin[100] founded on September 22, 2004, BitTorrent, Inc.[101] headquartered in San Francisco[102], California[103], as a privately held American[104] company[105] that develops transformative technology and products to continue the advancement of a more efficient and open Internet also promotes development of the BitTorrent protocol through R&D and open specifications.

BitTorrent (`http://www.bittorrent.com`) is also the name of the original implementation of the protocol it started as a Python[106] (source code and old versions[107]) application, now refereed as BitTorrent Mainline, to a full featured commercial enterprise. BitTorrent.com part of BitTorrent Inc. is now a destination to download entertainment content using the BitTorrent protocol. The site provide fast, on-demand access to the most comprehensive licensed catalog of thousands of movies, TV shows, music and games, but it also provides content creators a publishing platform to list their works in high-quality alongside the most recognizable titles from major movie studios, TV networks, and record labels.

97 `http://en.wikipedia.org/wiki/BitTorrent%20%28protocol%29`
98 `http://wiki.theory.org/BitTorrentSpecification`
99 `http://en.wikipedia.org/wiki/Bram%20Cohen`
100 `http://en.wikipedia.org/wiki/Ashwin%20Navin`
101 `http://en.wikipedia.org/wiki/BitTorrent%20%28company%29`
102 `http://en.wikipedia.org/wiki/San%20Francisco%2C%20California`
103 `http://en.wikipedia.org/wiki/California`
104 `http://en.wikipedia.org/wiki/United%20States`
105 `http://en.wikipedia.org/wiki/company`
106 `http://en.wikibooks.org/wiki/Subject%3APython_programming_language`
107 `http://download.bittorrent.com/dl/?C=M;O=D`

BitTorrent Inc. also contributes through more broadly focused standards bodies like the Internet Engineering Task Force (IETF) at the LEDBAT working group (`http://www.ietf.org/html.charters/ledbat-charter.html`). BitTorrent, Inc. owns the clients BitTorrent Mainline and µTorrent[108], as the BitTorrent DNA[109] (Delivery Network Accelerator) which is a free content delivery service based on the BitTorrent protocol that brings the power of user-contributed bandwidth to traditional content publishers while leaving publishers in full control of their files.

The BitTorrent protocol is peer-to-peer in nature, its innovative approach in the beginning, was due to not be centered about the creation a real distributed network but around the specific shared resources, in this case files, preferably large files, as users connect to each other directly to send and receive portions of a large file from other peers who have also downloaded either the file or parts of the it. These pieces are then reassembled into the full file. Since the users are downloading from each other and not from one central server, the bandwidth load of downloading large files is divided between the many sources that the user is downloading from. This decreases the bandwidth cost for people hosting large files, and increases the download speeds for the people downloading large files, because the protocol makes use of the upstream bandwidth of every downloader to increase the effectiveness of the distribution as a whole, and to gain advantage on the part of the downloader. However, there is a central server (called a tracker) which coordinates the action of all such peers. The tracker only manages connections, it does not have any knowledge of the contents of the files being distributed, and therefore a large number of users can be supported with relatively limited tracker bandwidth.

The key philosophy of BitTorrent is that users should upload (transmit outbound) at the same time they are downloading (receiving inbound.) In this manner, network bandwidth is utilized as efficiently as possible. BitTorrent is designed to work better as the number of people interested in a certain file increases, in contrast to other file transfer protocols.

BitTorrent is redefining the way people share and search for content and is getting very popular for downloading movies, TV shows, full music albums and applications (it gains in performance with other alternatives) since it is very file specific and it gains on the "new" factor of P2P content, more users equals more speed, but it will not be the optimum solution to rare files or to distribute content that is not highly sought over.

To download files that are hosted using BitTorrent users must have a BitTorrent client and to publish a file one must run a tracker.

In November 2004, BitTorrent accounted for an astounding 35 percent of all the traffic on the Internet and in 2006 the BitTorrent protocol has risen to over 60 percent of all Internet traffic according to British Web analysis firm CacheLogic. Due to this some ISPs are doing traffic shaping also know as bandwidth throttling, meaning they are reducing the protocol priority inside their networks and reducing its overall performance this has resulted in two kind of responses, some ISPs are investing in upgrading their networks and provide local cache to the protocol and implementors of the protocol are starting to battle ISPs that refuse to adapt by encrypting and randomizing it, this kick need to adapt and

108 `http://en.wikipedia.org/wiki/%CE%BCTorrent`
109 `http://en.wikipedia.org/wiki/BitTorrent%20DNA`

the increasing popularity due to deviation from its creators vision is placing more and more its evolution on the hands of independent developers.

The BitTorrent system is highly dependent on active participation of peers since it only goal is the sharing of files. Rare and "old" content is not easy to find on the system, only highly sought after content benefit from this P2P implementation. Small files also don't fully benefit from it, since the needed time for replication is too short and in some extreme situations can even degrade the experience.

2.15.1 Content indexers

There are many different BitTorrent websites that index content, each providing information about files distributed via the BitTorrent protocol. They typically contain multiple torrent files and an index of those files. In a typical scenario, a user would enter such a site and browse or search for the content they desire, based on the torrent descriptions posted at the site by other users. If a torrent with the sought content is found, the user could download that torrent.

Legal Torrents (`http://www.legaltorrents.com`), a collection of Creative Commons-licensed, legally downloadable, freely distributable creator-approved files, from electronic/indie music to movies and books, which have been made available via BitTorrent. Everyone that grabs the BitTorrent client and downloads helps contribute more bandwidth.

There is also OpenBitTorrent (`http://openbittorrent.com`), OpenBitTorrent.kg (`http://www.openbittorrent.kg`), myTorrentTracker (`http://www.mytorrenttracker.com`) and trackhub (`http://trackhub.appspot.com`), all BitTorrent trackers free for anyone to use to share files. You don't need to register, upload or index a torrent anywhere, all you have to do is to include the tracker URL in the torrent file.

Other:

- bt.etree.org (`http://bt.etree.org/`), a site provided by the etree.org community for sharing the live concert recordings of trade friendly artists.
- The Pirate Bay (`http://thepiratebay.org`).
- mininova (`http://www.mininova.org/`).

Such websites each have different features to facilitate the user's search. See Wikipedia's Comparison of BitTorrent sites[110] page. Several of the larger BitTorrent tracker sites were shut down citing concerns about problems with copyright holders, mostly representatives of large business interests. While in short it does prevent large scale copyright infringement, it also creates difficulty to legal uses and there is the issue with false notifications, that is claiming infringement of rights over works they do not own. In the long run this does little to solve the problem and pressures the protocol to evolve in ways to avoid this type of disruption. One way people have adapted to this pressure is to create private trackers that are only available by invitation.

110 `http://en.wikipedia.org/wiki/Comparison%20of%20BitTorrent%20sites`

2.15.2 Protocol

As already discussed the BitTorrent is a protocol for distributing files. It identifies content by URL and is designed to integrate seamlessly with the web. Its advantage over plain HTTP is that when multiple downloads of the same file happen concurrently, where each downloaders upload to each other, making it possible for the file source to support very large numbers of downloaders with only a modest increase in its load. (`http://www.bittorrent.com/protocol.html`). BitTorrent share some of the nomenclature of other P2P protocols but also creates new ones (see Wikipedia's page BitTorrent vocabulary[111] for an extended list).

Torrent

A *torrent* can mean either a `.torrent` metadata[112] file or all files described by it, depending on context. The **torrent file**[113], as defined in the BitTorrent specification, contains the URLs of multiple trackers[114], that coordinates communication between the peers in the swarm, and integrity metadata about all the files it makes down-loadable, including their names and sizes and checksums[115] of all pieces in the *torrent*. It can also contain additional metadata defined in extensions to the BitTorrent specification, known as *BitTorrent Enhancement Proposals*. Examples of such proposals include metadata for stating who created the torrent, and when.

Index[116]

An *index* is a list of .torrent files (usually including descriptions and other information) managed by a website[117] and available for searches. An *index* website can also be sometimes refereed as a *tracker*, but as a "Torrent" tracked not BitTorrent tracker).

The protocol was partially dependent of centralized in do to the requirement of trackers.

Client

The program that enables p2p[118] file sharing[119] via the BitTorrent protocol. It denotes still the semi-centralized nature of the protocol, on the protocol definition sometimes the terms is substituted by Peer (ie: peer_id[120]), for instance Gnutella refers to participants always as Peers or Nodes and to implementer as Vendor.

Tracker

A tracker[121] is a server that keeps track of which seeds and peers are in the swarm. Clients report information to the tracker periodically and in exchange receive information about

111 `http://en.wikipedia.org/wiki/BitTorrent%20vocabulary`
112 `http://en.wikipedia.org/wiki/metadata`
113 `http://en.wikipedia.org/wiki/Torrent%20file`
114 `http://en.wikipedia.org/wiki/BitTorrent%20tracker`
115 `http://en.wikipedia.org/wiki/Hash%20list`
116 `http://en.wikipedia.org/wiki/BitTorrent%20index`
117 `http://en.wikipedia.org/wiki/website`
118 `http://en.wikipedia.org/wiki/Peer-to-peer`
119 `http://en.wikipedia.org/wiki/file%20sharing`
120 `http://wiki.theory.org/BitTorrentSpecification#peer_id`
121 `http://en.wikipedia.org/wiki/BitTorrent%20tracker`

other clients to which they can connect. The tracker is not directly involved in the data transfer and does not have a copy of the file.

Scrape[122]

This is when a client sends a request to the tracking server for information about the statistics of the torrent, such as with whom to share the file and how well those other users are sharing.

With the adoption of DHT (Distributed Hash Tables) the BitTorrent protocol starts to become more that a semi-centralized distribution network around a single resource, it becomes more decentralized and removes the static point of control, the tracker, this is done by relying in DHTs and the use of the PEX[123] extension. Enabling the volatile Peer to operate also as a tracker, but even if this addressed the need for static tracker servers, there is still a centralization of the network around the content. Peers don't have any default ability to contact each other outside of that context.

Seeder

A *seeder* is a *client* that has a complete copy of the torrent and still offers it for upload. The more *seeders* there are, the better the chances of getting a higher download speed. If the seeder seeds the whole copy of the download they should get faster downloads

Seeding rules, like we will see with the special case of *super-seeding*, are variables and algorithms implemented locally by the client in a general configuration often open in some form to user control. These rules control and may serve to optimize the selection of what available torrent is seeded, instead of just starting the next one in the list, and sort torrents based on a **Seeding Rank**.

Seeding Rank

Is a priority rating, resulting from the calculations based on the active seeding rules of the client, serving to prioritizes your torrents based on how needy they are. It generates a priority queue, where the available torrents are given use of the available open slots for transfer. Several consideration can contribute to the *Seeding Rank*:

- Seed Ratio. The lower the ratio, the more scarce a the torrent is, the higher its Seeding Rank should be, giving priority to rare torrents.
- Seed Count. Similar to Seed ratio but considers not only complete seeds but the number of any client interested in the torrent, works in reverse, giving preference to larger swarms and torrents in high demand.
- Timed Rotation. Torrents will be rotated in and out of seeding mode. Each torrent is given a length of time they remain seeding.
- Default. Each torrents will be seeded based on their order they are added to the seed list.

Announce

Similar to "Scrape", but means that the client also announces that it wants to join the swarm and that the server should add it to the list of peers in that swarm.

122 `http://en.wikipedia.org/wiki/Tracker%20scrape`
123 Chapter 2.15.4 on page 63

Availability (also known as Distributed copies.)

This is a common word used on distributed systems, in this case it refers to the number of full copies of the file available to the client. Each seed adds 1.0 to this number, as they have one complete copy of the file. A connected peer with a fraction of the file available adds that fraction to the availability, if no other peer has this part of the file.

Example: a peer with 65.3% of the file downloaded increases the availability by 0.653. However, if two peers both have the same portion of the file downloaded - say 50% - and there is only one seeder, the availability is 1.5.

Interested

Describes a downloader who wishes to obtain pieces of a file the client has. For example, the uploading client would flag a downloading client as 'interested' if that client did not possess a piece that it did, and wished to obtain it.

Downloader

A *downloader* is any peer that does not have the entire file and is downloading the file. This term, used in Bram Cohen's[124] Python[125] implementation, lacks the negative connotation attributed to *leech*. Bram chose the term *downloader* over *leech*[126] because BitTorrent's tit-for-tat ensures downloaders also upload and thus do not unfairly qualify as *leeches*.

Choked

Describes a client that has been refused file pieces. A client *chokes* another client in several situations:

- The second client is a *seed*, in which case it does not want any pieces (i.e., it is completely *uninterested*)
- The client is already uploading at its full capacity (it has reached the value of `max_uploads`)
- The second client has been blacklisted[127] for being abusive or is using a blacklisted BitTorrent client.

Snubbed

An uploading client is flagged as *snubbed* if the downloading client has not received any data from it in over 60 seconds.

2.15.3 Extension protocol for BitTorrent

Created by Arvid Norberg and Ludvig Strigeus, (description `http://www.rasterbar.com/products/libtorrent/extension_protocol.html`), it is an extension to the protocol that intends to provide a simple and thin transport for future extensions to the BitTorrent protocol. This protocol makes it easy to add new extensions without interfering with the standard bittorrent protocol or clients that don't support this extension.

124 `http://en.wikipedia.org/wiki/Bram%20Cohen`
125 `http://en.wikipedia.org/wiki/Python%20%28programming%20language%29`
126 `http://en.wikipedia.org/wiki/Leech%20%28computing%29`
127 `http://en.wikipedia.org/wiki/Blacklist%20%28computing%29`

The extension messages IDs are defined in the handshake is to avoid having a global registry of message IDs. Instead the names of the extension messages requires unique names, which is much easier to do without a global registry.

There seems also to be a concurrent implementation, or variation, by Vuze (`http://wiki.vuze.com/w/Azureus_messaging_protocol`) that is used both by Vuse and Transmission.

2.15.4 Peer exchange

Peer exchange[128] or **PEX** is a communications protocol[129] that augments the BitTorrent[130] file sharing protocol. It allows a group of users (or peers[131]) that are collaborating to share a given file to do so more swiftly and efficiently. PEX is implemented using one of two common extension protocols.

In the original design of the BitTorrent file sharing protocol clients, that users (Peers) in a file sharing group (known as a "swarm") relied upon a central computer server called a tracker[132] to find each other and to maintain the swarm. PEX greatly reduces the reliance of peers on a tracker by allowing each peer to directly update others in the swarm as to which peers are currently in the swarm. By reducing dependency on a centralized tracker, PEX increases the speed, efficiency, and robustness of the BitTorrent protocol making it more decentralized.

As already explained, users wishing to obtain a copy of a file typically first download a .torrent[133] file that describes the file(s) to be shared, as well as the URLs[134] of one or more central computers called trackers[135] that maintain a list of peers currently sharing the file(s) described in the .torrent file. In the original BitTorrent design, peers then depended on this central tracker to find each other and maintain the swarm. Later development of distributed hash table[136]s (DHTs) meant that partial lists of peers could be held by other computers in the swarm and the load on the central tracker computer could be reduced. PEX allows peers in a swarm to exchange information about the swarm directly without asking (polling[137]) a tracker computer or a DHT. By doing so, PEX leverages the knowledge of peers that a user is connected to by asking them for the addresses of peers that they are connected to. This is faster and more efficient than relying solely on a tracker and reduces the processing load on the tracker. It also keep swarms together when the tracker is down. In fact removing any control over the distribution once a peer keeps a complete copy of the file share.

Peer exchange cannot be used on its own to introduce a new peer to a swarm. To make initial contact with a swarm, each peer must either connect to a tracker using a ".torrent" file, or else use a router computer called a bootstrap node[138] to find a distributed hash table

128 `http://en.wikipedia.org/wiki/Peer%20exchange`
129 `http://en.wikipedia.org/wiki/communications%20protocol`
130 `http://en.wikipedia.org/wiki/BitTorrent%20%28protocol%29`
131 `http://en.wikipedia.org/wiki/Peer-to-peer`
132 `http://en.wikipedia.org/wiki/BitTorrent%20tracker`
133 `http://en.wikipedia.org/wiki/Torrent_file`
134 `http://en.wikipedia.org/wiki/URL`
135 `http://en.wikipedia.org/wiki/BitTorrent%20tracker`
136 `http://en.wikipedia.org/wiki/distributed%20hash%20table`
137 `http://en.wikipedia.org/wiki/Polling%20%28computer%20science%29`
138 `http://en.wikipedia.org/wiki/bootstrap%20node`

(DHT) which describes a swarm's list of peers. For most BitTorrent users, DHT and PEX will start to work automatically after the user launches a BitTorrent client and opens a .torrent file. A notable exception is "private torrents" which are not freely available; these will disable DHT.

It was agreed between the Azureus and µTorrent developers that any clients which implement either of the mechanisms above try to obey the following limits when sending PEX messages:

- There should be no more than 50 added peers and 50 removed peers sent in any given PEX message.
- A peer exchange message should not be sent more frequently than once a minute.

Some clients may choose to enforce these limits and drop connections from clients that ignore these limits.

2.15.5 Permanent DHT tracking

With the PEX implementation and reliance on the distributed hash table (DHT), the evolution into creating a real P2P overlay network that is completely serverless was the next logical stem, much like the eDonkey network has evolved as we have seen. The DHT works mostly the same way and will take information not only from old trackers by also from the PEX implementation, creating something like a distributed Database of shared torrents acting as backup tracker when all other trackers are down or can't deliver enough peers, as well as enabling trackerless torrents. The DHT acts and is added to torrents as a pseudo-tracker if the client has the option enabled and DHT trackers can be enabled and disabled per torrent just like regular trackers. Clients using this permanent DHT tracking are now a fully connected decentralized P2P network, they enter the DHT as a new node, this of course makes it necessary for private trackers (or non-public distributions) to exclude themselves from the participating.

Bootstrapping the DHT

Since the DHT is independent of any single tracker (and point of failure), the issue of how the DHT routing table is bootstraped, the first time using DHT, has to be addressed. This is done in several ways:

1. Manually entering a host name and port number of a DHT node.
2. Connect to a tracker that has a .torrent file with a list of DHT nodes.
3. Downloading any torrent with peers who advertise that they support DHT. Not fully supported by all clients as it requires clients to advertise in the BitTorrent handshake DHT support.

Magnet links

Traditionally, .torrent files are downloaded from torrent sites. But several clients also support the Magnet URI scheme[139]. A magnet link can provide not only the torrent hash needed to seek the needed nodes sharing the file in the DHT but may include a tracker for the file.

139 `http://en.wikipedia.org/wiki/Magnet%20URI%20scheme`

2.15.6 BitTorrent Enhancement Proposal (BEP)

The BitTorrent Enhancement Proposal Process (BEP) is a process started by John Hoffman. The process is defined in the public domain document (`http://www.bittorrent.org/beps/bep_0001.html`).

List of BitTorrent Enhancement Proposals is available (`http://www.bittorrent.org/beps/bep_0000.html`).

A BEP is a design document providing information to the BitTorrent community, or describing a new feature for the BitTorrent protocols. BEPs should provide a concise technical specification of the feature and a rationale for the feature, and are intended to be the primary mechanisms for proposing new features, for collecting community input on an issue, and for documenting the design decisions that have gone into BitTorrent.

The BEP author is responsible for building consensus within the community and documenting dissenting opinions. Because the BEPs are maintained as re-structured text files in a versioned repository, their revision history is the historical record of the feature proposal.

2.15.7 Super-seeding

Super-seeding[140] specified in BEP 16 (`http://www.bittorrent.org/beps/bep_0016.html`) is a extension to the BitTorrent protocol[141] (implemented without changes to the protocol), intended to be used when there is only one seed[142], it permits to manage the scarcity of a resource.

In a situation when a seeder detects that it is the only source for a file, it will attempt to minimize the amount of data uploaded, as to guarantee external sharing and optimize access to the scarce resource, until it detects that other complete seeders exist. The feature was conceived by John Hoffman and first implemented in the BitTornado[143] client in 2003.

2.15.8 uTP

uTP or **µTP** (sometimes also referred as **Micro Transport Protocol**[144]) is an open source[145] cross-platform[146] protocol[147], created to be implemented on top of UDP[148] protocol a TCP-like implementation of LEDBAT[149] (a TCP congestion avoidance algorithm[150]).

140 `http://en.wikipedia.org/wiki/Super-seeding`
141 `http://en.wikipedia.org/wiki/BitTorrent%20%28protocol%29`
142 `http://en.wikipedia.org/wiki/Terminology%20of%20BitTorrent`
143 `http://en.wikipedia.org/wiki/BitTornado`
144 `http://en.wikipedia.org/wiki/Micro%20Transport%20Protocol`
145 `http://en.wikipedia.org/wiki/open%20source`
146 `http://en.wikipedia.org/wiki/cross-platform`
147 `http://en.wikipedia.org/wiki/Protocol%20%28computing%29`
148 `http://en.wikipedia.org/wiki/User%20Datagram%20Protocol`
149 `http://datatracker.ietf.org/wg/ledbat/charter/`
150 `http://en.wikipedia.org/wiki/TCP%20congestion%20avoidance%20algorithm`

µTP was developed within BitTorrent, Inc.[151] with no input from either the networking or BitTorrent communities as to be provide reliable, ordered delivery while maintaining minimum extra delay as too automatically slow down the rate at which packets of data are transmitted between users of peer-to-peer[152] file sharing torrents[153] when it interferes with other applications. For example, the protocol should automatically allow the sharing of an ADSL line between a BitTorrent application and a web browser. It was first introduced in the µTorrent 1.8.x beta branches, and publicized in the alpha builds of µTorrent[154] 1.9. as is now the primary transport protocol for uTorrent peer-to-peer connections.

uTP is documented as a BitTorrent extension, in BEP 29 (`http://bittorrent.org/beps/bep_0029.html`). BitTorrent, Inc. has made a C++ implementation available under the MIT license (`http://github.com/bittorrent/libutp`), but the external interface is strictly C (ANSI C89).

2.15.9 BitTorrent protocol encryption

As of January 2005, BitTorrent traffic made up more than a third of total residential Internet traffic. Some ISPs decided to take different measures control and event to subvert P2P traffic, as covered in Shadow play[155] section of this book.

This created a need for providing a BitTorrent protocol encryption[156]. Obfuscation and encryption makes traffic harder to detect and monitor and therefore harder to throttle. BitTorrent protocol encryption is not designed to provide anonymity[157] or confidentiality[158], even if some solutions will it increase confidentiality by obfuscating the content.

Bram Cohen, the inventor of BitTorrent[159], opposed adding encryption to the BitTorrent protocol. Cohen stated he was worried that encryption could create incompatibility between clients, stressing the point that the majority of ISPs don't block the torrent protocol. Cohen wrote "I rather suspect that some developer has gotten rate limited by his ISP, and is more interested in trying to hack around his ISP's limitations than in the performance of the Internet as a whole". After some criticism for this position, Cohen later added the ability to receive but not originate encrypted connections on his Mainline client[160]. Notably, when µTorrent was purchased by BitTorrent, Inc. and then became the next mainline release, the ability to originate encrypted connections was retained, but it became turned off by default.

Encryption will not stop a traffic shaping system configured to universally slow down all encrypted, unidentifiable or unknown protocols using a method as simple as packet loss. Encrypting tracker communications prevents eavesdropping on peer lists and does not require

151 `http://en.wikipedia.org/wiki/BitTorrent%20%28company%29`
152 `http://en.wikipedia.org/wiki/peer-to-peer`
153 `http://en.wikipedia.org/wiki/BitTorrent%20%28protocol%29`
154 `http://en.wikipedia.org/wiki/%C2%B5Torrent`
155 Chapter 1.5.2 on page 27
156 `http://en.wikipedia.org/wiki/BitTorrent%20protocol%20encryption`
157 `http://en.wikipedia.org/wiki/anonymity`
158 `http://en.wikipedia.org/wiki/confidentiality`
159 `http://en.wikipedia.org/wiki/BitTorrent%20%28protocol%29`
160 `http://en.wikipedia.org/wiki/BitTorrent%20%28software%29`

upgrading both ends of peer-to-peer connections, but it requires imposing computational overhead on the tracker.

Protocol header encryption (PHE)

Created by RnySmile and first implemented in BitComet version 0.60 on 8 September 2005. The specifications was neither published, nor is it compatible with MSE/PE, and there are claims that it was already reverse engineered, reducing its usefulness.

Message stream encryption (MSE)/Protocol encryption (PE)

Developed by Azureus (now Vuze) in late January 2006, that later suffered several alterations permitting a broader acceptance by the creators of other BitTorrent clients.

As per the specifications (`http://wiki.vuze.com/w/Message_Stream_Encryption`), MSE/PE uses key exchange[161] combined with the infohash of the torrent to establish an RC4[162] encryption key. The key exchange helps to minimize the risk of passive listeners, and the infohash helps avoid man-in-the-middle attack[163]s. RC4 is chosen for its speed. The first kilobyte of the RC4 output is discarded to prevent a particular attack[164].

The specification allows for the users to choose between encrypting the headers only or the full connection. Encrypting the full connection provides more obfuscation but uses more CPU time. To ensure compatibility with other clients that don't support this specification, users may also choose whether unencrypted incoming or outgoing connections are still allowed. Supported clients propagate the fact that they have MSE/PE enabled through PEX[165] and DHT[166].

Analysis of this method has shown that statistical measurements of packet sizes and packet directions of the first 100 packets in a TCP session can be used to identify the obfuscated protocol with over 96% accuracy, this makes this solution only effective against the efforts of ISPs that don't adopt state of the art traffic analysis, mostly smaller ISPs.

Various solutions exist to protect the BitTorrent network against attacks including encrypting both peer-to-tracker and peer-to-peer communication, using Microsoft's Teredo[167] so that TCP connections are tunneled within UDP packets, filtering TCP resets before they reach the TCP layer in the end-host, or switching entirely from a TCP-based transport to a UDP-based transport. Each solution has its trade-offs. Filtering out attack TCP resets typically requires kernel access, and the participation of the remote peer since the attacker has to send the reset packet to the local and remote peers. Teredo is not available on all BitTorrent clients. Rewriting TCP reliability, in-order delivery and congestion control in a new UDP protocol represents a substantial engineering effort and would require upgrading both ends of any peer-to-peer connection.

161 `http://en.wikipedia.org/wiki/Diffie-Hellman%20key%20exchange`
162 `http://en.wikipedia.org/wiki/RC4`
163 `http://en.wikipedia.org/wiki/man-in-the-middle%20attack`
164 `http://en.wikipedia.org/wiki/RC4%23Fluhrer%2C%20Mantin%20and%20Shamir%20attack`
165 `http://en.wikipedia.org/wiki/peer%20exchange`
166 `http://en.wikipedia.org/wiki/Distributed%20hash%20table`
167 `http://en.wikipedia.org/wiki/Teredo%20tunneling`

2.15.10 Software Implementations

Wikipedia provides an some relevant information in articles like comparison of BitTorrent software[168] and usage share of BitTorrent clients[169]. The following list is given as to provide a general idea and comparison regarding the implementation details in relation to other P2P protocols.

(This should not to be considered a complete list of BitTorrent clients, no special order was used. All links have been verified, special attention has been given to the programming language and licenses of the software. Last update 11 September 2010)

- **BitTorrent Queue Manager** (`http://btqueue.sourceforge.net`), a console-based BitTorrent Client with built-in scheduler for handling multiple sessions. It is designed to manage sessions in queue easily without heavy-weight GUI. External module can search for new torrents in trackers and submit it automatically. OpenSource (Python Software Foundation License) project, using Python.
- **Vuze**[170] previously called Azureus (`http://azureus.sourceforge.net` or `http://www.vuze.com/`), an open source BitTorrent client in Java, probably the more advanced peer for the network (multiple torrent downloads, queuing/priority systems, start/stop seeding options, embedded tracker, Mainline DHT and a lot more) but a known resource hog, consuming large quantities of memory and CPU power.
- **µTorrent**[171] (`http://utorrent.com`), a closed source, freeware BitTorrent client in C++, a very complete peer (includes bandwidth prioritization, scheduling, RSS auto-downloading and Mainline DHT and more) with a very low system footprint.
- **BitTornado**[172] (`http://bittornado.com`), an open source BitTorrent client in Python based on the original BitTorrent client.
- **BitComet**[173] (`http://www.bitcomet.com`), (originally named SimpleBT client from versions 0.11 to 0.37) is a closed source but freeware, BitTorrent client for the MS Windows OS only, it also supports HTTP/FTP download management.
- **ABC (Yet Another BitTorrent Client)**[174] (`http://pingpong-abc.sourceforge.net`), an open source BitTorrent client, based on BitTornado.
- **Transmission** (`http://transmission.m0k.org/`), an open source lightweight BitTorrent client with a simple graphic user interface on top of a cross-platform back-end. Transmission runs on Mac OS X with a Cocoa interface, Linux/NetBSD/FreeBSD/OpenBSD with a GTK+ interface, and BeOS with a native interface. Released under the MIT/X Consortium License.
- **Warez** (`http://www.warezclient.com`), a closed source, MS Windows only BitTorrent client from Neoteric Ltd. (previously supporting the Ares Network Warez P2P client).
- **Bits on Wheels** (`http://bitsonwheels.com`), a freeware but closed source, implementation, written in Objective-C and Cocoa for the Macintosh.

168 `http://en.wikipedia.org/wiki/Comparison%20of%20BitTorrent%20software`
169 `http://en.wikipedia.org/wiki/Usage%20share%20of%20BitTorrent%20clients`
170 `http://en.wikipedia.org/wiki/Vuze`
171 `http://en.wikipedia.org/wiki/%CE%BCTorrent`
172 `http://en.wikipedia.org/wiki/BitTornado`
173 `http://en.wikipedia.org/wiki/BitComet`
174 `http://en.wikipedia.org/wiki/ABC%20%28Yet%20Another%20BitTorrent%20Client%29`

- **Vidora** (`http://www.videora.com/`), a closed source, freeware implementation that also support Really Simple Syndication (RSS) feeds.
- **sharktorrent** (`http://sharktorrent.sourceforge.net/`), an open source (GNU GPL) written in C++. This a cross-platform BitTorrent client uses QT , libtorrent and boost libraries.
- **ted [Torrent Episode Downloader]** (`http://www.ted.nu/`), an open source (GNU GPL) BitTorrent client coded in Java, it also support torrent RSS feeds.
- **[175]** (`http://libtorrent.rakshasa.no`) is a text-based ncurses[176] BitTorrent client written in C++, based on the libTorrent libraries for Unix (not to be confused with libtorrent by Arvid Norberg/Rasterbar), whose author's goal is "a focus on high performance and good code. Both the client as the library are available under the GNU GPL.
- **libtorrent[177]** (`http://www.rasterbar.com/products/libtorrent/`) from Rasterbar Software, an open source C++ library, implementing the BitTorrent protocol and core necessities for an application, using zlib and Boost libraries, specifically Boost.Asio and shares the Boost license. This library is also commonly used embedded in devices[178]. The library also provides support for UPnP configuration.
  - **Halite** (`http://www.binarynotions.com/halite-bittorrent-client`), an open-source, under the Boost Software License, this BitTorrent client uses the libtorrent library. Coded in C++ using the Boost library and WTL (Windows only).
  - **FireTorrent** (`https://addons.mozilla.org/en-US/firefox/addon/10931`) by Pete Collins, Radical Software Ltd, Jan Varga, Matthew Gertner, open source in JavaScript using the libtorrent library, Mozilla Public License FireFox extension/add-on to download torrents.
  - **Folx** (`http://www.mac-downloader.com/`), closed source, using libtorrrent library (Mac only).
  - **qBittorrent[179]** (`http://www.qbittorrent.org/`), open source GNU GPL, developed by a Ph.D student (Christophe Dumez), a Bittorrent client using C++ / libtorrent and a Qt4 Graphical User Interface.
  - **Deluge[180]** (`http://deluge-torrent.org`), an open source, using Python and libtorrent, lightweight, cross-platform BitTorrent client in Python released under the GNU GPL license.
  - Limewire[181], already covered as a well known Gnutella implementation also support the BitTorrent protocol by using the libtorrrent library.
  - **BTG** (`http://btg.berlios.de`), Bittorrent client implemented in C++ and using the Rasterbar Libtorrent library, released under the GNU GPL. Provides a Ncurses, SDL, Gtkmm and WWW GUI, which communicate with a common backend running the actual BitTorrent operation, available only for OSX, BSD and Linux.

175 `http://en.wikipedia.org/wiki/rTorrent`
176 `http://en.wikipedia.org/wiki/ncurses`
177 `http://en.wikipedia.org/wiki/libtorrent%20%28Rasterbar%29`
178 Chapter 3.8.9 on page 118
179 `http://en.wikipedia.org/wiki/qBittorrent`
180 `http://en.wikipedia.org/wiki/Deluge%20%28software%29`
181 Chapter 2.11.3 on page 51

- **Free Download Manager**[182] (FDM), (`http://www.freedownloadmanager.org`), C++ open source software distributed under GNU GPL using the libtorrrent library (Windows only).
- **torrent2exe.com**, a web tool, that reportedly converts .torrents into executables (Windows) for distribution, closed source using the libtorrrent library.
- **Flush** (`http://sourceforge.net/projects/flush`) GTK-based BitTorrent client for Linux, open source C++/GTK+ using the libtorrrent library.
- **Pump** (`http://www.vipeers.com`), a closed source video manager that supports the BitTorrent protocol by using the libtorrrent library.
- **Lince** (`http://lincetorrent.sourceforge.net`), open source C++/GTK+/libtorrrent BitTorrent client, release under the GNU GPL (Linux/BSD/UNIX-like OSes).
- **Miro**[183], previously known as Democracy Player and DTV (`http://getmiro.com`) is an designed to automatically download videos from RSS-based "channels", manage them and play them. Open source in Python/GTK/libtorrrent, released under the terms of the GNU General Public License.
- **tvitty** (`http://tvitty.com`) closed source BitTorrent download add-in for vista media center using libtorrrent (Windows only).
- **FatRat** (`http://fatrat.dolezel.info`), is an open source download manager for Linux written in C++ using Qt4 and libtorrrent libraries.
- **LeechCraft** (`http://leechcraft.org`), open source BitTorrent client (supporting also HTTP/FTP downloads), create using C++, Qt and libtorrrent. Released under the GNU General Public License.
- **MooPolice** (`http://www.moopolice.de`), BitTorrent client for Windows, having an unorthodox GUI. Open source (without a specific license) C++ using MFC and libtorrent BitTorrent client library and MiniUPnP.
- **Linkage** (`http://code.google.com/p/linkage`), a lightweight BitTorrent client written in C++ using gtkmm and libtorrent, open source under the GNU General Public License (no longer maintained).
- **Arctic Torrent** (`http://int64.org/projects/arctic-torrent`), a small BitTorrent client for Windows (includes a 64bits version). Open Source C++ under the MIT License, using libtorrent.

2.15.11 Specific BitTorrent Papers

- May 22, 2003 - Incentives Build Robustness in BitTorrent (PDF)[184], Bram Cohen
  The BitTorrent file distribution system uses tit-for-tat as a method of seeking pareto efficiency. It achieves a higher level of robustness and resource utilization than any currently known cooperative technique. We explain what BitTorrent does, and how economic methods are used to achieve that goal.

182 `http://en.wikipedia.org/wiki/Free%20Download%20Manager`
183 `http://en.wikipedia.org/wiki/Miro%20%28software%29`
184 `http://web.archive.org/web/20060320112940/http://www.bittorrent.com/bittorrentecon.pdf`

2.16 Other Software Implementations

2.16.1 JXTA

JXTA™ technology, created by Sun™ (`http://www.jxta.org`), is a set of open protocols that allow any connected device on the network ranging from cell phones and wireless PDAs to PCs and servers to communicate and collaborate in a P2P manner. JXTA peers create a virtual network where any peer may interact with other and their resources directly even when some of the peers and resources are behind firewalls and NATs or are on different network transports. The project goals are interoperability across different peer-to-peer systems and communities, platform independence, multiple/diverse languages, systems, and networks, and ubiquity: every device with a digital heartbeat. The technology is licensed using the Apache Software License (similar to the BSD license).

Most of the implementation is done in Java (with some minor examples in C).

2.16.2 iFolder

iFolder[185] (`http://www.ifolder.com`) is an still in early development open source[186] application, developed by Novell, Inc.[187], intended to allow cross-platform[188] file sharing[189] across computer networks by using the Mono/.Net framework.

iFolder operates on the concept of shared folders[190], where a folder is marked as shared and the contents of the folder are then synchronized to other computers over a network, either directly between computers in a peer-to-peer[191] fashion or through a server. This is intended to allow a single user to synchronize their files[192] between different computers (for example between a work computer and a home computer) or share files with other users (for example a group of people who are collaborating on a project).

The core of the iFolder is actually a project called Simias[193]. It is Simias which actually monitors files for changes, synchronizes these changes and controls the access permissions on folders. The actual iFolder clients (including a graphical desktop client and a web client) are developed as separate programs that communicate with the Simias back-end.

The iFolder client runs in two operating modes, enterprise sharing (with a server) and workgroup sharing (peer-to-peer, or without a server).

185 `http://en.wikipedia.org/wiki/iFolder`
186 `http://en.wikipedia.org/wiki/open%20source`
187 `http://en.wikipedia.org/wiki/Novell%2C%20Inc.`
188 `http://en.wikipedia.org/wiki/cross-platform`
189 `http://en.wikipedia.org/wiki/file%20sharing`
190 `http://en.wikipedia.org/wiki/Directory%20%28file%20systems%29`
191 `http://en.wikipedia.org/wiki/peer-to-peer`
192 `http://en.wikipedia.org/wiki/File%20synchronization`
193 `http://en.wikipedia.org/wiki/Simias%20%28software%29`

2.16.3 Freenet

The Freenet Project (`http://freenetproject.org`), designed to allow the free exchange of information over the Internet without fear of censorship, or reprisal. To achieve this Freenet[194] makes it very difficult for adversaries to reveal the identity, either of the person publishing, or downloading content. The Freenet project started in 1999, released Freenet 0.1 in March 2000, and has been under active development ever since.

Freenet is unique in that it handles the storage of content, meaning that if necessary users can upload content to Freenet and then disconnect. We've discovered that this is a key requirement for many Freenet users. Once uploaded, content is mirrored and moved around the Freenet network, making it very difficult to trace, or to destroy. Content will remain in Freenet for as long as people are retrieving it, although Freenet makes no guarantee that content will be stored indefinitely.

The journey towards Freenet 0.7 began in 2005 with the realization that some of Freenet's most vulnerable users needed to hide the fact that they were using Freenet, not just what they were doing with it. The result of this realization was a ground-up redesign and rewrite of Freenet, adding a "darknet" capability, allowing users to limit who their Freenet software would communicate with to trusted friends. This would make it far more difficult for a third-party to determine who is using Freenet.

Freenet 0.7 also embodies significant improvements to almost every other aspect of Freenet, including efficiency, security, and usability. Freenet is available for Windows, Linux, and OSX. It can be downloaded from:

Software Implementations

All software is available on The Freenet Project page.

Frost an application for Freenet that provides usenet-like message boards and file uploading/downloading/sharing functionalities. It should get installed with Freenet 0.7 automatically if you used the standard Freenet installers.

jSite is a graphical application that you can use to create, insert and manage your own Freenet sites. It was written in Java by Bombe.

Thaw is a filesharing utility and upload/download manager. It is used as a graphical interface for Freenet filesharing.

2.16.4 KaZaa

KaZaa (`http://www.kazaa.com`)

Software (FastTrack) Implementations

- **Kazaa**

194 `http://en.wikipedia.org/wiki/Freenet`

- **Kazaa Lite**
- **Diet Kaza**
- **giFT**
- **Grokster**
- **iMesh**

2.16.5 GNUnet

GNUnet (`http://gnunet.org/`), was started in late 2001, as a framework for secure peer-to-peer networking that does not use any centralized or otherwise trusted services. The framework provides a transport abstraction layer and can currently encapsulate the network traffic in UDP (IPv4 and IPv6), TCP (IPv4 and IPv6), HTTP, or SMTP messages. All peer-to-peer messages in the network are confidential and authenticated.

The primary service build on top of the framework is anonymous file sharing, implemented on top of the networking layer allows anonymous censorship-resistant file-sharing. GNUnet uses a simple, excess-based economic model to allocate resources. Peers in GNUnet monitor each others behavior with respect to resource usage; peers that contribute to the network are rewarded with better service.

GNUnet is part of the GNU project. Our official GNU website can be found at (`http://www.gnu.org/software/gnunet/`), there is only an existing client, OpenSource, GPL, written in C, that shares the same name as the network. GNUnet can be downloaded from this site or the GNU mirrors.

2.16.6 MANOLITO (MP2P)

MANOLITO or **MP2P** is the internal protocol name for the proprietary peer-to-peer file sharing network developed by Pablo Soto[195]. MANOLITO uses UDP connections on port 41170 for search routing and is based on Gnutella. In addition file transfers use a proprietary protocol based on TCP.

MANOLITO hosts obtain an entry into the network by contacting an HTTP network gateway, which returns a list of approximately one-hundred MANOLITO hosts. Hosts can also be manually connected to. Servents maintain contact with a fixed number of peers (depending on the Internet connection) that are sent search queries and results.

Software Implementations

- Blubster[196] (`http://www.blubster.com`), first implementation of the MP2P protocol, closed source but freeware for Windows.
- Piolet[197] (`http://www.piolet.com`), closed source but freeware for Windows.

195 `http://en.wikipedia.org/wiki/Pablo%20Soto`
196 `http://en.wikipedia.org/wiki/Blubster`
197 `http://en.wikipedia.org/wiki/Piolet`

2.16.7 Mute File Sharing

MUTE File Sharing (`http://mute-net.sourceforge.net`) is an anonymous, decentralized search-and-download file sharing system. MUTE uses algorithms inspired by ant behavior to route all messages, include file transfers, through a mesh network of neighbor connections.

Author Jason Rohrer - jcr13 (at) cornell (dot) edu Created using C++ and Crypto++ Library, support is provided for multiple OSs there is a frontend for Windows created with MFC, Mute is Open Source and released under the GPL License.

2.16.8 iMesh

iMesh[198] (`http://www.imesh.com`), a free but closed source P2P network (IM2Net) operating on ports 80, 443 and 1863, for Widows. iMesh is owned by an American company iMesh, Inc. and maintains a development center in Israel. An agreement with the MPAA had also been reached. Video files more than 50mb in size and 15 minutes in length can no longer be shared on the iMesh network, guaranteeing feature-length releases cannot be transferred across the network.

2.16.9 BitCoop

BitCoop (`http://bitcoop.sourceforge.net/)` created by Philippe Marchesseault is a console (Text Based) peer to peer backup system that enables the storage of files on remote computers with cryto and compression support. The size of files depends on the quantity you wish to share with the other peers. It is intended for server farms that wish to backup data among themselves. Supports various Operating Systems including Windows, Linux and Mac OS X, it's implemented in Java (Open Source under the GPL).

2.16.10 CSpace

CSpace (`http://cspace.in/)` provides a platform for secure, decentralized, user-to-user communication over the Internet. The driving idea behind the CSpace platform is to provide a connect(user,service) primitive, similar to the sockets API connect(ip,port). Applications built on top of CSpace can simply invoke connect(user,service) to establish a connection. The CSpace platform will take care of locating the user and creating a secure, nat/firewall friendly connection. Thus the application developers are relieved of the burden of connection establishment, and can focus on the application-level logic! CSpace is developed in Python. It uses OpenSSL for crypto, and Qt for the GUI. CSpace is licensed under the GPL.

198 `http://en.wikipedia.org/wiki/iMesh`

2.16.11 I2P

I2P is a generic anonymous and secure peer to peer communication layer. It is a network that sits on top of another network (in this case, it sits on top of the Internet). It is responsible for delivering a message anonymously and securely to another location.

2.16.12 p300

p300 (`http://p300.eu/`) is a P2P application created in Java with the intention of provide a just-works-single-download solution for a multitude of Operating Systems without the need to deal with user accounts or specific protocol and security configurations (ie. samba). Another aspect is that p300 is primarily meant to be used in LANs or over VPNs. This application is OpenSource released under the GNU GPL v3.

2.16.13 Netsukuku

Netsukuku (`http://netsukuku.freaknet.org/`) is a p2p (in mesh) network system, originally developed by FreakNet MediaLab, that can generate and sustain itself autonomously. It is designed to handle an unlimited number of nodes with minimal CPU and memory resources. It seems that it can be easily used to build a worldwide distributed, anonymous decentralized network, above the Internet, without the support of any servers, ISPs or authority controls. Netsukuku replaces the network level 3 of the OSI model[199] with another routing[200] protocol. An open source Python implementation was finished in October 2007.

Netsukuku is based on a very simple idea: extending the concept of Wi-Fi[201] mesh network[202]s to a global scale, although not necessarily using that medium. With the use of specialized routing protocols and algorithms, the current Wi-Fi technologies can be exploited to allow the formation of a global P2P wireless network, where every peer (node) is connected to its neighbors.

Other media will be equally functional to interconnect nodes, as the interaction is independent of that which it is transmitted through, but it is believed that Wi-Fi will be the most practical for ordinary users to take advantage of. Once greater proliferation has been achieved, it may become common to see some nodes establishing high-speed land-line connections between each other in the interest of increasing network bandwidth for connections over it and lowering latencies.

2.16.14 Adobe's RTMFP (Real Time Media Flow Protocol) Groups

The RTMFP[203] is a closed protocol/implementation based on the creation of Amicima[204], a start-up founded in 2004 focused in the development of improved Internet protocols for

199 `http://en.wikipedia.org/wiki/OSI%20model`
200 `http://en.wikipedia.org/wiki/routing`
201 `http://en.wikipedia.org/wiki/Wi-Fi`
202 `http://en.wikipedia.org/wiki/mesh%20network`
203 `http://en.wikipedia.org/wiki/Real%20Time%20Media%20Flow%20Protocol`
204 `http://en.wikipedia.org/wiki/Amicima`

client-server and peer-to-peer networking and derived applications (p2p-hackers - amicima's MFP - preannouncement, Jul 2005[205], MFP - The Secure Media Flow Protocol - version 1[206]), that was acquired by Adobe[207] for inclusion into the Flash platform, that gives developers the ability to stream data to endpoints without going through a central server (Flash Media Server[208]). This addition to the Flash player v10.1+ capabilities enables most P2P network needs to be performed on Flash. No much information is available on the implementation yet. The presentation is available in a flash video (`http://tv.adobe.com/watch/max-2009-develop/p2p-on-the-flash-platform-with-rtmfp`).

2.16.15 Other

- **Thunderbolt (aka Thunder)** (`http://www.xunlei.com/`) was created by Xunlei[209] Network Technology Ltd. Thunderbolt proprietary P2P network supports Multi-protocol P2P resources (supports BitTorrent, eDonkey, Kad, and FTP) and also HTTP downloads (a download accelerator) as it web caches to aid in accelerating of downloads. It is mainly used in the Mainland China, recently an English translation has been released. Of particularly interest is that on January 5, 2007, Google acquired a 4% stake on the company.
- **Cassandra** (`http://code.google.com/p/the-cassandra-project/`) a Structured Storage System on a P2P Network for managing structured data while providing reliability at a massive scale. Made in Java under the Apache License 2.0.
- **Aimini** (`http://www.aimini.com/),` windows freeware.
- **TinyP2P** (`http://www.freedom-to-tinker.com/tinyp2p.html`), claims to be *The World's Smallest P2P Application*, written in fifteen lines of code, in the Python programming language.
- Secure Content Downloader[210], closed source, Windows and Microsoft specific P2P application.
- **XNap** (`http://xnap.sourceforge.net/`) OpenSource (GPL), written in Java. The client features a modern Swing based user interface and console support. Able to work in several P2P Networks OpenNap, Gnutella, Overnet and OpenFT (and other networks supported by giFT like FastTrack). It also supports ICQ and IRC, viewers for MP3 tags, images, PDF, ZIP files and Text-To-Speech.
- Filetopia[211] (`http://www.filetopia.com`), a free but closed source server/clients P2P application for Windows. It includes, instant messaging, chat and file sharing system with a search engine, online friends list and message boards. It also supports the use of a bouncer (open source,Java) as an anonymity layer, that enables indirect connections.

205 http://zgp.org/pipermail/p2p-hackers/2005-July/002801.html
206 http://web.archive.org/web/20060906221358/www.amicima.com/downloads/documentation/protocol-doc-20051216.txt
207 http://en.wikipedia.org/wiki/Adobe%20Systems
208 http://en.wikipedia.org/wiki/Adobe%20Flash%20Media%20Server
209 http://en.wikipedia.org/wiki/Xunlei
210 http://www.microsoft.com/downloads/details.aspx?familyid=9a927cf6-16e4-4e21-9608-77f06d2156bb&displayla
en
211 http://en.wikipedia.org/wiki/Filetopia

- **Carracho** (`http://www.carracho.com`), freeware but closed sourced P2P application to MacOS X.
- **SockeToome** (`http://www.blackdiamond.co.za/bdsock.html`), shareware P2P application for Mac and Windows.
- **FilePhile** (`http://www.filephile.net`), closed source but freeware, Java-based and so multi-platform application.
- Napster network
  - WinMX
  - Napigator
  - FileNavigator
- WPNP network
  - WinMX
- other networks
  - MojoNation
  - Carracho
  - Hotwire
  - Chord peer-to-peer lookup service|Chord
  - Dexter
  - Swarmcast
  - Alpine program|Alpine
  - Scribe
  - Groove
  - Squid_Soft|Squid
  - Akamai
  - Evernet
  - Overnet network
  - Audiogalaxy network
  - SongSpy network
  - The Circle
  - OpenFT
- Acquisition
- Cabos
- Swapper
- SoulSeek

212

212 `http://en.wikibooks.org/wiki/Category%3AThe%20World%20of%20Peer-to-Peer%20%28P2P%29`

3 Building a P2P System

3.1 Developer

There are many reasons why one should invest time and effort in creating a P2P Application. As we already covered in the first chapter the P2P technology touches and has begun even to have impact in many fields of human endeavor. Besides the normal reasons behind any type of programming task (money, fame, and fun), it can also be an expression of a political stance or a vehicle to implement new concepts in networks or even economics. The bottom line is that P2P resumes itself to person-to-person, people working together for a specific goal. It is up to you, the programmer to create the infrastructure necessary to make this happen.

Any P2P project aims to become a widely used, trusted and reliable. Few are open, secure, free, non-discriminating, egalitarian, unfettered and censorship-resistant.

3.1.1 Selecting the Programming Language

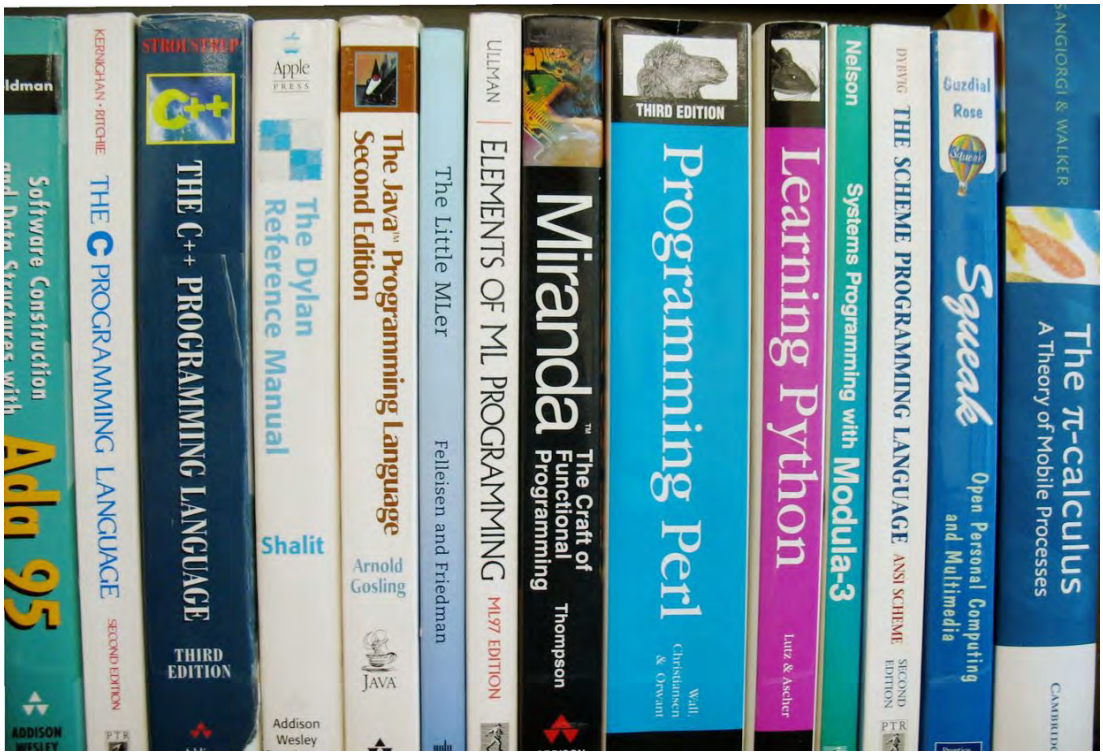

Figure 7 Programming Languages

3.1.2 Selecting the License

Selecting the right license is the most important step for any software for public distribution. It will determine how the project will be done, even restrict the programming language that is selected to implement the solution, the time it takes to get to a final product and is of increasingly importance to the relation with the users.

There is a never ending list of licenses one can use out of the box and as the implementer you can even create your own. On this last step be very careful if you don't have a grasp on all the implications you shouldn't risk it.

The Open Source Initiative (OSI) offers a great an annotated definition of what Open Source is, here (`http://www.opensource.org/docs/osd`).

Open vs Closed Source

Figure 8

3.1.3 How can P2P generate revenue

Figure 9

"New Rule No. 1: Get profitable sooner. Forget B2B or B2C: The new catchphrase among dot-coms is P2P, for path to profitability. Venture-capital investors once tolerated profits forecasts three or four years in the future. Now they're looking for business plans that target profitability less than one year after an initial public offering."
—Dennis K. Berman, "Dot Coms: Can They Climb Back?," Business Week, June 26, 2000

Donations

Donations is a model that is open for all types of software, open source or closed source, the objective is to let users to freely contribute to a project they like, most probably you will not get a fixed income of this revenue source but it may not be the only way you use to get a payoff or even profit from the project. If you use this method attempt to be clear on how the donations will be used (to further development, etc...), immediate needs you may have (hosting, services and equipment to develop and test).

Depending were your project is located and how it is structured and focused there are ways to maximize the revenue, a common way to run a donation based project is to set up a non-profit corporation (you can even issue receipts for tax purposes).

Most people my not like it but providing a donors/supporters page list does incentive participation, and may event show users that small amounts are of help, if you decide to list donors do offer an option to be excluded from it.

For an in depth look on the most of donation problems, you may read the article, **When Do Users Donate?** Experiments with Donationware: Ethical Software, Work Equalization, Temporary Licenses, Collective Bargaining, and Microdonations (`http://www.donationcoder.com/Articles/One/index.html`). You may even try to join their project or support similar ones, like for instance microPledge (`http://micropledge.com/`) or even create a similar offering around your own product.

Examples:

- **Software in the Public Interest(SPI)**[1] (`http://www.spi-inc.org/`) a non-profit organization which was founded to help organizations develop and distribute open hardware and software. (used by the Debian project)

Money

Hardware

It is also common programmers or projects to accept hardware donations by request or as incentive to the project for adding support to exclusive features or specific setups. If you adopt to support this feature, do provide and maintain a list of hardware that is wanted and how it would help you.

Shareware / for Pay

This is the most problematic setup due to the legal hot-waters it can get you into and the formalisms and obligations that you need to comply to.

Also restricting the participation on the network will be intentionally reducing its usefulness, this is why most P2P services are free or at least support some level of free access.

Variations
There are several models that are variations of the simple donation/pay model, they give specific goals to the users or to the project in relation to the values collected.

Ransom

Put features or the code of the application up for a ransom payment, if people do contribute and fill that goal you accept to comply with your proposal (ie: opening the source code, fix or implement a feature).
Ransom has been shown to works in practice, it is used in several open source initiatives and even writers have tested this scheme, an example of the later is the test the writer Lawrence Watt-Evans has done on several titles[2], all successfully reached his monetary and production goal.

Pay for features

In a variation of the ransom model, in this particular case you should be extra careful to inform users on what they are paying for, and the legality of what you are providing for that payment. Extra feature may be better services or even a better quality for the existing ones.

Paid support

Paid support include providing users access to a paid prioritized service for technical support, this is very commonly use on Open Source projects. You should restrain yourself for over complicating the software so you can profit from it, as the users will be the network. One solution is to provide a default dumb down version for public consumption and enable

1 `http://en.wikipedia.org/wiki/Software%20in%20the%20Public%20Interest`
2 `http://www.ethshar.com/thesprigganexperiment0.html`

a very high degree of tweaking of the software, protocol or network and then attempt to profit for it.

License new technology

In case you came up with a new technology or a way new interconnect existing ones that to can be made into revenue source.

Venture Capital

Level of Control

3.2 The Peer

The Peer and the user running it, is the corner stone of all P2P systems, without peers you will not be able to create the Network, this seems obvious but it is very common to disregard the users needs and focus on the final objective, the Network itself, kind of looking to a forest and not seeing the trees.

3.2.1 Building communities

There is a benefit in incorporating the social element into the software. Enabling people to come together not only to share content or services but around a common goal were cooperation for an optimal state should remain the final objective.

Figure 10

A P2P application is a type of social software

Social software is defined as any software that promotes and enables social collaboration. This is of course part of what defines a modern p2p system, where the gatekeeper is no more and participants are free to interact on their own terms. The p2p applications becomes an enabling tool in this reality.

The world can increasingly be defined as a network of networks, all things are interconnected at some level. As ubiquity of Internet becomes a reality it not only makes communications speed increase but also the volume of sources this creates a quality assurance problem.

Online collaboration, built around the shared goal of a functional peer to peer network will help not only to improve the system but to establish a relation of trust that is followed by the emergence of a reputation latter. Across the participants and in the network and software

itself, this personalization will ultimately enable to extend this trust and reputation to the multitude of relations that are possible in a peer to peer network.

Social Networks (person-to-person)

As communities emerge and aggregate, they increase the addressability not only of content as before but between producers and consumers that will self organize as to establishing of super and subgroups (many-to-many) based on personal preferences.

Enabling meaningful exchange of information and promoting increased collaboration also increases disponibility of rare or obscure content and the importance of microcontent, due to the removal for extraneous information, since content can consistently be directed to the right audience (pulled, not pushed).

Using a social software or any type of centralized social network permits the easy extraction of metrics on user actions this has been highly explored by corporations, that have for some time been waging war not only to get a share of the data such interactions generate but also for control of such information (ie: Google and Microsoft). Microsoft has a research project dedicated especially to this subject (Enterprise Social Computing[3]).

Opening the walled gardens

3.2.2 A user oriented GUI

As you start to project your P2P application the GUI[4] is what the users will have to interact with to use your creation, you should attempt to define not only what OS[5] you will support but within what framework you can design the application to be used from a WEB browser or select a portable framework so you can port it to other systems.

Become the user

Overwhelming a user with options is always a bad solution and will only be enticing to highly experienced users, even if it done based on what you like, you should keep in mind that you aren't creating it for your own use.

Simplicity is the ultimate sophistication

Aside form the technical decisions the functionally you offer should also be considered, the best approach is to be consistent and offer similar options to existing implementations, other applications or even how it is normally done on the environment/OS you are using. There are several guidelines you may opt to follow for instance Apple provides a guideline for OSX (`http://developer.apple.com/documentation/UserExperience/Conceptual/OSXHIGuidelines/`).

Content is king

The difference is in the little details

3 `http://research.microsoft.com/en-us/projects/EnterpriseSocComp/`
4 `http://en.wikipedia.org/wiki/Graphical%20user%20interface`
5 `http://en.wikipedia.org/wiki/Operating%20system`

Guide the user

3.3 P2P Networks Topology

The topology of a P2P network can be very diverse, it depends on the medium it is run (Hardware), the size of the network (LAN,WAN) or even on the software/protocol that can impose or enable a specific network organization to emerge.

Below we can see the most common used topologies (there can be mixed topologies or even layered on the same network), most if not all peer to peer networks are classified as overlay networks[6], since they are created over an already existing network with its own topology.

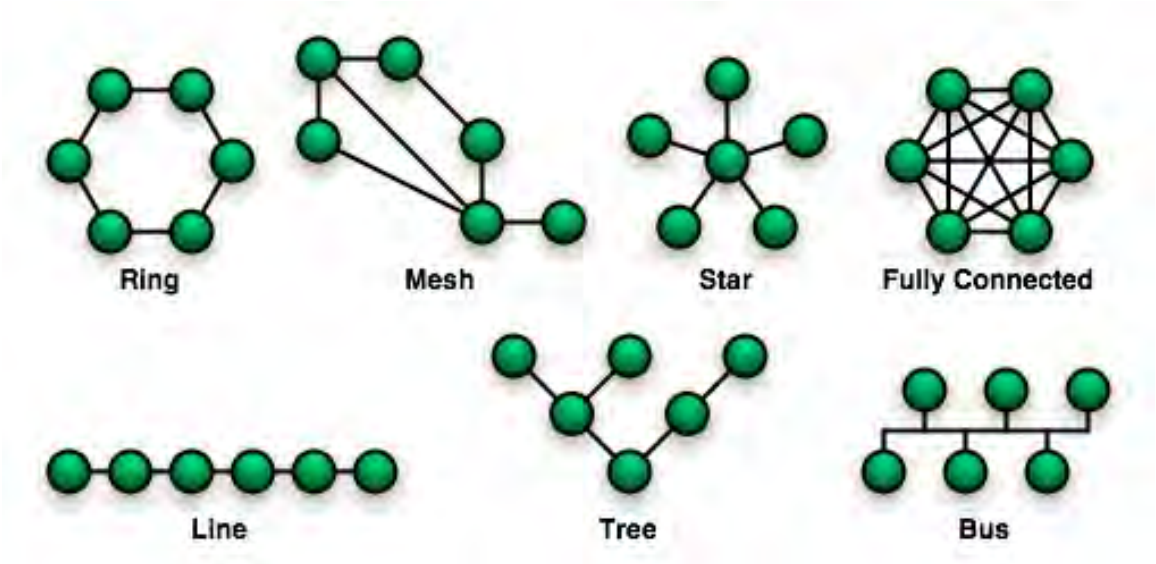

Figure 11

Ring topology[7] - Mesh[8]topology - Star topology[9] - Total Mesh or **Full Mesh[10] -**Line topology**-**Tree topology[11]**- Bus topology[12] - Hybrid[13]topology**

A fully connected P2P network is not feasible when there is a large numbers of peers participating.

- In network of several thousands or millions of participants (n = participants)
- Were a peer would have to handle O(n^2) overall connections, it wouldn't scale.

6 http://en.wikipedia.org/wiki/Overlay%20network
7 http://en.wikipedia.org/wiki/Ring%20topology
8 http://en.wikipedia.org/wiki/Mesh%20network
9 http://en.wikipedia.org/wiki/Star%20topology
10 http://en.wikipedia.org/wiki/Mesh%20network
11 http://en.wikipedia.org/wiki/Tree%20topology
12 http://en.wikipedia.org/wiki/Bus%20topology
13 http://en.wikipedia.org/wiki/hybrid

Since most P2P networks are also a type of overlay networks, the resulting topology of a P2P system depends also on the protocol, the infrastructure (medium/base network) or by the interaction of peers. This "variables" add more layers to the complexity of the normal networks, were the basic characteristics are bandwidth, latency and robustness, making most P2P networks into self-organizing overlay networks.

When performing studies of P2P networks the resulting topology (structural properties) is of primary importance, there are several papers on the optimization or characteristics of P2P networks.

The paper *Effective networks for real-time distributed processing* (`http://arxiv.org/abs/physics/0612134`) by Gonzalo Travieso and Luciano da Fontoura Costa, seems to indicate that uniformly random interconnectivity scheme, is specific Erdős-Rényi[14] (ER) random network model with fixed number of edges, as being largely more efficient than the scale-free counterpart, the Barabási-Albert[15](BA) scale-free model.

3.3.1 Testing/Debugging the system

Virtual Machines

Software Tools

3.3.2 Self-Organizing Systems (SOS)

P2P systems fall by default in this category, taking in account that most try to remove the centralization or Control and Command (C&C) structure that exists in most centralized systems/networks.

The study of self-organizing systems is relatively new, and it applies to a huge variety of systems or structures from organizations to natural occurring events). Mostly done with math and depending on models and simulations, its accuracy depends on the complexity of the structure, number of intervenients and initial options.

This book will not cover this aspect of the P2P systems in great detail but some references and structural characteristics do run in parallel with the SOS concepts. For more information on SOS check out the Self-Organizing Systems (SOS) FAQ for USENET Newsgroup comp.theory.self-org-sys (`http://www.calresco.org/sos/sosfaq.htm`).

Synergy
One of the emerging characteristic of SOS is synergy, where members aggregate around a common goal each working to the benefit of all. This is also present in most P2P systems were peers work to optimally share resources and improve the network.

Swarm
The effect generated of participants in a SOS, to aggregate and do complementary work so to share knowledge and behavior in a way it improves coordination. This is observable in

14 `http://en.wikipedia.org/wiki/Erd%C5%91s-R%C3%A9nyi%20model`
15 `http://en.wikipedia.org/wiki/BA%20model`

the natural world on flocks of birds or social insects. We will covert this aspect of P2P later on, but keep it in mind that it related to SOS.

3.3.3 Bootstrap

Most P2P systems don't have (or need) a central server, but need to know an entry point into the network. This is what is called bootstrapping the P2P application. It deals with the connectivity of peers, to be able find and connect other P2P peers (the network) even without having a concrete idea who and what is where.

This is not a new problem; it is shared across several network technologies, that avoid the central point of failure of requiring a central server to index all participants - a gatekeeper.

One solution, Zero Configuration Networking (zeroconf)[16] (see `http://www.zeroconf.org/`), compromises a series of techniques that automatically (without manual operator intervention or special configuration servers) creates a usable Internet Protocol (IP) network. These techniques are often used to help bootstrap, configure or open a path across routers and firewalls.

3.3.4 Hybrid vs real-Peer systems

One of the main objectives of the P2P system is to make sure no single part of it critical to the collective objective. By introducing any type of centralization to a peer-to-peer Network one is creating points of failure, as some Peers will be more than others this can even lead to security or stability problems, as with the old server-client model, were a single user could crash the server and deny its use to others.

3.3.5 Availability

Due to the instability of a P2P network, where nodes are always joining and leaving, some efforts must be made to guarantee not only that the network is available and enabling new peers to join, but that the resources shared continue to be recognized or at least indexed while temporarily not available.

3.3.6 Integrity

Due to the open nature of peer-to-peer networks, most are under constant attack by people with a variety of motives. Most attacks can be defeated or controlled by careful design of the peer-to-peer network and through the use of encryption. P2P network defense is in fact closely related to the "Byzantine Generals Problem[17]". However, almost any network will fail when the majority of the peers are trying to damage it, and many protocols may be rendered impotent by far fewer numbers.

16 `http://en.wikipedia.org/wiki/Zero%20configuration%20networking`
17 `http://en.wikipedia.org/wiki/Byzantine%20fault%20tolerance%23The%20dilemma`

3.3.7 Clustering

Computer science defines a **computer cluster**[18] in general terms as a group of tightly coupled computer[19]s that work together closely so that in many respects they can be viewed as though they are a single computer.

The components of a cluster are commonly, connected to each other through fast networks and usually deployed to improve performance and/or availability over that provided by a single computer, while typically being much more cost-effective than single computers of comparable speed or availability.

As we have seen before, this concept if applied to distributed networks or WANs (in place of LANs), generates distributed computations, grids and other systems. All of those applications are part of the P2P concept.

We loosely define clusters as a physical, social or even economical/statistical event. That is is defined by the aggregation of entities due to sharing a property in common, that property may be a shared purpose or a characteristic, or any other communality.

As we look at the topologies generated by P2P networks we can observe that most protocols generate some kind of clustering around networks structures, resources and they can even emerge as a result of the status of network conditions. Clustering is then an unsupervised learning problem, an automatic emerging event that results on the creation of ad hoc collection of unlabeled objects (data/items or events). For more information on clusters you may check *A Tutorial on Clustering Algorithms* (`http://home.dei.polimi.it//matteucc/Clustering/tutorial_html/`).

A cluster is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters using the same set of characteristics.

This concept is very important on not only the form of P2P networks but has also implications on the social structure/relations that can be build upon the use of P2P applications.

Flashcrowds

Flashcrowd is a behavioral model in that participants will tend to aggregate/crowd around an event, in P2P terms this can be a scarce resource, for instance as we will see later BitTorrent promotes big files over small ones and new over old, this is a result of a connection to the network based on single items, the speeds on BitTorrent does depend uppermost on generating flashcrowds around files (more peers, more speed that will snow ball in more seeders).

This can also result in a DoS (Denial of Service) or flood. For instance, if a P2P system is poorly designed, attempting to connect a significant number of peers to the network may disturb the bootstrap method used.

You should educate users, more connections doesn't equate to more speed, at most it will result in more responses to queries but that may depend on how the protocol is structured,

18 `http://en.wikipedia.org/wiki/Computer%20cluster`
19 `http://en.wikibooks.org/wiki/computer`

but enabling them will cost bandwidth. On the other side more peers will result on more resources to be shared that will also result in overlapping of shares and so better speeds, as a P2P network gets bigger the better it can provide for its users.

Optimizations

There are simple optimizations that should be done in any P2P Protocol that could bring a benefit to both peers and the network in general. These are based on system metrics or profile like IP (ISP or range), content, physical location, share history, ratios, searches and many other variables.

Most of the logic/characteristics of P2P networks and topologies (in a WAN environment) will result in aggregation of peers and so this clusters will share the same properties of Distributed Behavioral Models, like Flocks, Herds and Schools this results in an easier to study environment and to establish correlation about the peers relations and extrapolate ways to improve efficiency.

- Alignment

  To get information/characteristics from the local system/environment in order to optimize the peer "location" on the network by selecting and optimize the separation and cohesion functions to improve the local neighborhood.

- Separation

  To implement a way to avoid crowding with other peers based on unwanted alignment.

- Cohesion

  To select peers based on their own alignment.

3.4 P2P Networks Traffic

Figure 12

The P2P traffic detected on the Internet due to the nature of the protocols and topology used some times can only be done by estimation based on the perceived use (users on-line, number of downloads of a given implementation) or by doing point checks on the networks itself. It is even possible to access the information if the implementation on the protocol or application was done with this objective in mind, several implementations of Gnutella a for example have that option and Bearshare did even reports some of the users system parameters, like type of firewall etc...

Examples of services that provide such traffic information over P2P networks are for instance Cachelogic (`http://www.cachelogic.com/research/2005_slide16.php#`).

Communication

One of your most important considerations is how you project the way peers will communicate, even if we discard the use of central servers like SuperNodes and multiple distributed clients as Peer/Nodes there will be several questions to consider:

- Will communication need to go across firewalls and proxy servers?
- Is the network transmission speed important? Can it be configured by the user?
- Will communications be synchronous or asynchronous?
- Will it need/use only a single port? what port should we use?
- What resources will you need to support? Is there size limits? Should compression be used ?
- Will the data have to be encrypted?
- etc...

One must carefully consider your project's specific goals and requirements, this will help you evaluate the use of toolkits and frameworks. Try not to reinvent the wheel if you can't came up with a better solution or have the time,capacity or disposition to. You can also use open standards (ie: use the HTTP protocol) to, but are also free to explore other approaches.

Today even small LANs will have at least a firewall and probably a router even if all components are under the user's control, some user will just lack the knowledge on how to set them properly. Another consideration is on the simple configuration requirements of the application, most users will have problems dealing with technical terms and dependencies, even new versions of the Windows OS will have a default enabled firewall, those may prove to be a unsurmountable barrier for users, thankfully there are some tools available to make life easier for all.

3.4.1 UPnP

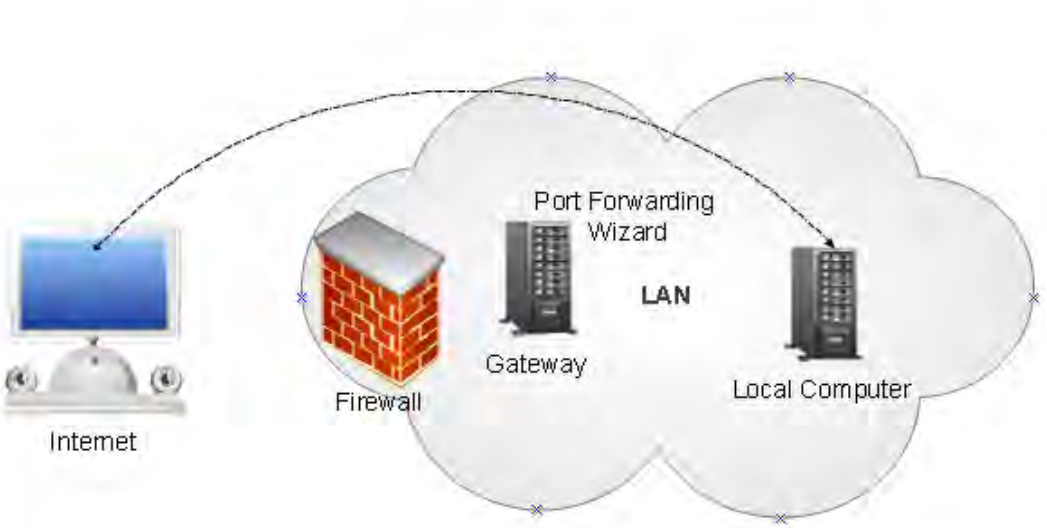

Figure 13 Internet Firewall and LAN with port forwarding.

The Universal Plug and Play (UPnP)[20] architecture consists in a set of open standards and technologies promulgated by the UPnP Forum[21] (`http://www.upnp.org/`), with the goal of extending the Plug and Play concept to support networks and peer-to-peer discovery (automatic discovered over the network. wired or wireless), configuration and control and so enable that appliances, PCs, and services be able to connect transparently. Permitting any UPnP device to dynamically join a network, obtain its IP address and synchronize capabilities (learn from and inform other devices). It can also be seen in general terms as similar to a distributed Simple Network Management Protocol (SNMP)[22].

As UPnP is offered in most modern routers, network devices and it is also supported by Microsoft since Windows XP. While supposedly aimed to address the problem for network programs users in accepting incoming connection from the Internet ("port forwarding" or "NAT traversal"), as it would remove the necessary step in configuring router to accept incoming connections and then route them to the LAN's local machine behind the router, something that is hard to explain and for the common user to understand.

All this makes it a necessity that a Windows P2P application should support this architecture programmatically so to avoid the requirements that users deal with the necessary changes when UPnP is enabled (by default is should be disable due to security risks). With the added necessity that if there is a firewall on the local machine that doesn't conform to UPnP, it must have its configuration changed so to enable the necessary TCP (port 2859) and UDP (port 1900) communications for UPnP.

20 `http://en.wikipedia.org/wiki/Universal%20Plug%20and%20Play`
21 `http://en.wikipedia.org/wiki/UPnP%20Forum`
22 `http://en.wikipedia.org/wiki/Simple%20Network%20Management%20Protocol`

Microsoft provides the UPnP Control Point API. It is available in Windows Me, CE .Net, XP and later in the system services "SSDP Discovery Service" (ssdpsrv) and "Universal Plug and Play Device Host" (upnphost) or by COM libraries. It can be used in C++ or Visual Basic applications or in scripts embedded in HTML pages.

Further information on the UPnP technology on Windows can be gathered on this sources:

- Programming control point application using the UPnP Control Point API (`http://www.codeproject.com/KB/IP/upnplib.aspx`) by amatecki
- Using UPnP for Programmatic Port Forwardings and NAT Traversal (`http://www.codeproject.com/KB/IP/PortForward.aspx`) by By Mike O'Neill

Freely usable implementation

- CyberLink (`http://sourceforge.net/projects/clinkcc)` for C++ is a development package for UPnP programmers. Using the package, you can create UPnP devices and control points easily. released under the BSD License.
- MiniUPnP Project (`http://miniupnp.free.fr`) open source C implementation under a BSD compatible license.
- GUPnP (`http://www.gupnp.org`), an object-oriented open source framework for creating UPnP devices and control points, written in C using GObject and libsoup. It provides the same set of features as libupnp, but shields the developer from most of UPnP's internals. Released under the GNU LGPL.
- UPNPLib (`http://www.sbbi.net/site/upnp`) open source Java implementation under a Apache Software License.

3.4.2 Security Considerations

By using a P2P system users will broadcast their existence to others, this in contrast to a centralized service were they may interact with others but their anonymity can be protected.

Violating the security of a network can be a crime, for instance the 2008 case of the research project from University of Colorado and University of Washington[23], the researchers engaged in the motorization of users across the Tor anonymous proxy network and could have faced legal risks for the snooping.

This vulnerability of distributed communications can result in identity attacks (e.g. tracking down the users of the network and harassing or legally attacking them), DoS, Spamming, eavedropping and other threats or abuses. All this actions are generally targeted to a single user and some may even be automated, there are several actions the creator can take to make it more difficult but ultimately they can't be stopped and should be expected and dealt with, one of the first steps is to provide information to the user so they can locally implement hardware or software actions and even a social behavior to counteract this abuse.

23 `http://news.cnet.com/8301-13739_3-9997273-46.html`

DoS (denial of service), Spamming

Since each user is a "server" they are also prone to denial of service[24] attacks (attacks that may, if optimized, make the network run very slowly or break completely), the result may depend on the attacker resources and how the decentralized is the P2P protocol on the other hand to be the target of spamm[25] (e.g. sending unsolicited information across the network- not necessarily as a denial of service attack) does only depend how visible and contactable you are, if for instance other users can send messages to you. Most P2P applications support some kind of chat system and this type of abuse is very hold on such system, they can address the problem but will complete solve it, what can lead to social engineering attacks were users can be lead to perform actions that will compromise them or their system, on this last point only giving information to users that enables them to be aware of the risk will work.

ARP Attack

Eavesdropping

3.4.3 Hardware traffic control

3.4.4 Software traffic control

Since most Network applications and in specific P2P tools are prone to be a source of security problems (they will bypass some of the default security measures from inside), when using or creating such a tool one must take care on granting the possibility or configuring the system to be as safe as possible.

Tools for security

There are several tools and options that can be used for this effect, be it configuring a firewall, adding a IP blocker or making sure some restrictions are turned on by default as you deploy your application.

- PeerGuardian 2 (`http://phoenixlabs.org/pg2/`) a OpenSource tool produced by Phoenix Labs', consisting in a IP blocker for Windows OS that supports multiple lists, list editing, automatic updates, and blocking all of IPv4 (TCP, UDP, ICMP, etc)
- PeerGuardian Lite (`http://phoenixlabs.org/pglite/`) a version of the PeerGuardian 2 that is aimed at having a low system footprint.

Blocklists

Blocklists are text files containing the IP addresses of organizations opposed and actively working against file-sharing (such as the RIAA), any enterprise that mines the networks or

24 `http://en.wikipedia.org/wiki/Denial-of-service%20attack`
25 `http://en.wikipedia.org/wiki/spamming`

attempts to use resources without participating in the actual sharing of files. It is basically a spam filter like the ones that exist for eMail systems.

3.4.5 Firewalls

As computers attempt to be more secure for the user, todays OS will provide by default some form of external communication restriction that will permit the user to define different levels of trust, this is called a Firewall[26]. A Firewall may have hardware[27] or software[28] implementation and is configured to permit, deny, or proxy[29] data[30] through a computer network[31]. Most recent OSs will come with a software implementation running, since a connection to the Internet are becoming common and the lack or even the default configuration of the Firewall can cause some difficulties to the use of P2P applications.

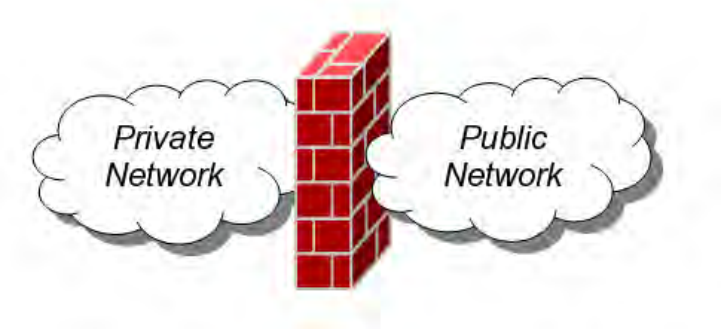

Figure 14

on Windows

Microsoft in the last OS releases has taken the option for the security of user but without his intervention to include and enabling by default a simple firewall solution since Windows XP SP2. This blocks any incoming (HTTP over port 80 or mail over ports 110 or 25), the classification of "unsolicited messages" is a bit over empathized since the messages can well be part of a P2P (or any other type of distributed) network. This can also have implications on blocking UPnP capabilities on the local machine.

- ICF (Internet Connection Firewall)[32] is included with Microsoft's Windows XP, Windows Server 2003, and Windows Vista operating systems.

26 http://en.wikipedia.org/wiki/Firewall%20%28networking%29
27 http://en.wikipedia.org/wiki/hardware
28 http://en.wikipedia.org/wiki/software
29 http://en.wikipedia.org/wiki/Proxy%20server
30 http://en.wikipedia.org/wiki/data
31 http://en.wikipedia.org/wiki/computer%20network
32 http://en.wikipedia.org/wiki/Windows%20Firewall

3.4.6 Multi-cast

3.4.7 Routers

Routing

NAT

NAT[33] (network address translation[34])

NAT Traversal

Users behind NAT should be able to connect with each other, there are some solutions available that try to enable it.

STUN

STUN (*Simple Traversal of UDP over NATs*) is a network protocol[35] which helps many types of software and hardware receive UDP[36] data properly through home broadband router[37]s that use *NAT*).

Quoted from its standard document, RFC 3489[38]:

> *"Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs) (STUN) is a lightweight protocol that allows applications to discover the presence and types of NATs and firewalls[39] between them and the public Internet[40].*
>
> It also provides the ability for applications to determine the public IP address[41] allocated to them by the *NAT*.
>
> *"STUN works with many existing NATs, and does not require any special behavior from them. As a result, it allows a wide variety of applications to work through existing NAT infrastructure."*

As *STUN* RFC states this protocol is not a cure-all for the problems associated with *NAT* but it is particularly helpful for getting voice over IP working through home routers. VoIP[42] signaling protocols like SIP[43] use UDP packets for the transfer of sound data over the Internet, but these UDP packets often have trouble getting through *NATs* in home routers.

33 http://en.wikipedia.org/wiki/NAT
34 http://en.wikipedia.org/wiki/network%20address%20translation
35 http://en.wikipedia.org/wiki/network%20protocol
36 http://en.wikipedia.org/wiki/User%20Datagram%20Protocol
37 http://en.wikipedia.org/wiki/router
38 http://www.faqs.org/rfcs/rfc3489.html
39 http://en.wikipedia.org/wiki/firewall%20%28networking%29
40 http://en.wikipedia.org/wiki/Internet
41 http://en.wikipedia.org/wiki/IP%20address
42 http://en.wikipedia.org/wiki/VoIP
43 http://en.wikipedia.org/wiki/SIP

STUN is a client-server[44] protocol. A VoIP phone or software package may include a *STUN* client, which will send a request to a STUN server. The server then reports back to the STUN client what the public IP address of the *NAT* router is, and what port was opened by the *NAT* to allow incoming traffic back in to the network.

The response also allows the *STUN* client to determine what type of *NAT* is in use, as different types of *NATs* handle incoming UDP packets differently. It will work with three of four main types: full cone NAT[45], restricted cone NAT[46], and port restricted cone NAT[47]. It will not work with symmetric NAT[48] (also known as bi-directional *NAT*) which is often found in the networks of large companies.

44 http://en.wikipedia.org/wiki/client-server
45 http://en.wikipedia.org/wiki/full%20cone%20NAT
46 http://en.wikipedia.org/wiki/restricted%20cone%20NAT
47 http://en.wikipedia.org/wiki/port%20restricted%20cone%20NAT
48 http://en.wikipedia.org/wiki/symmetric%20NAT

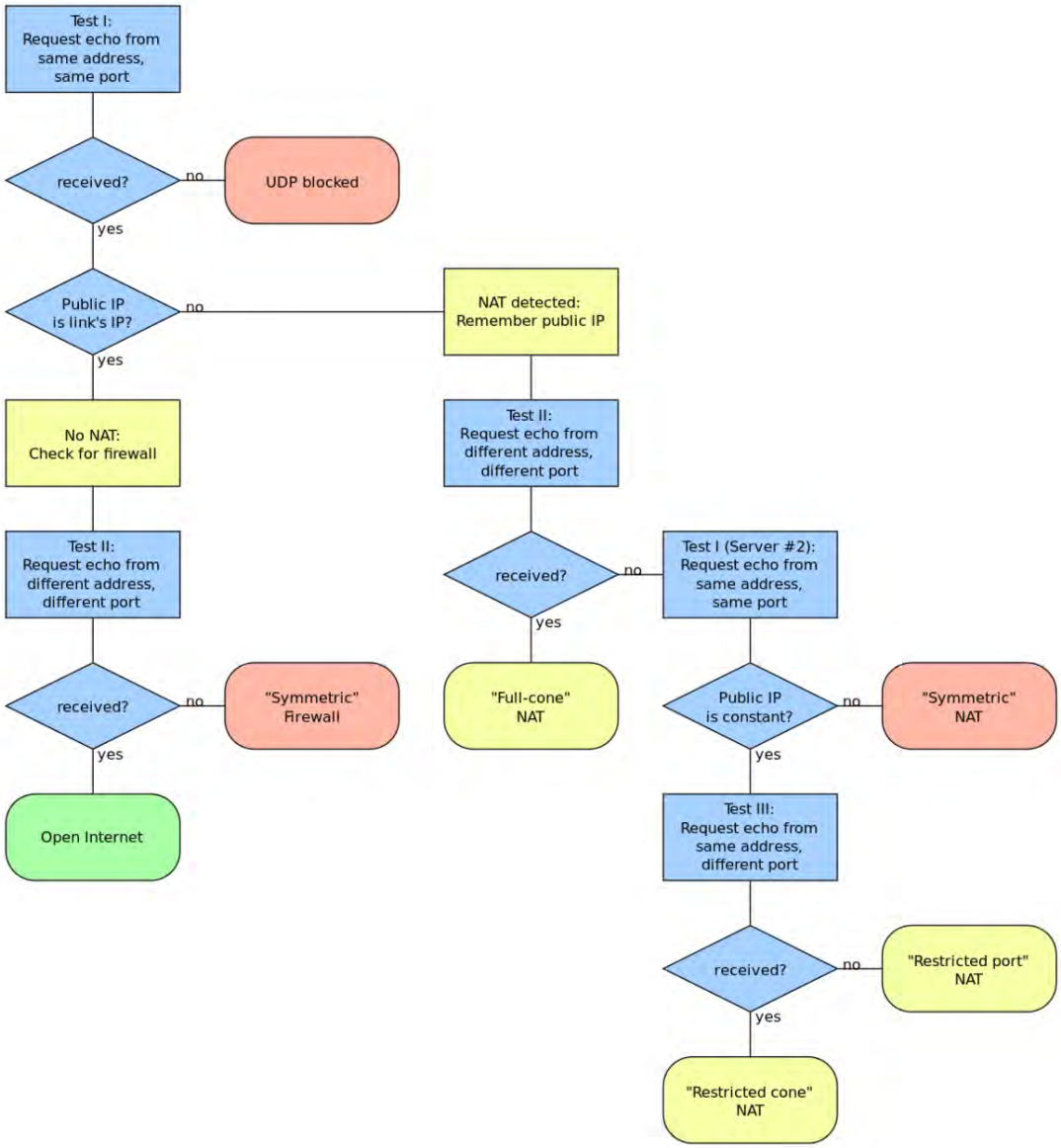

Figure 15 Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)) algorithm.

Port Forwarding

If a peer is behind a router or firewall (using NAT) it may be necessary to configure it by hand, as to allow the P2P programs to function properly, this can be a daunting task for those technically challenged and can even have an impact in the adoption of the P2P application. This can be resolved automatically using some solutions like UPnP.

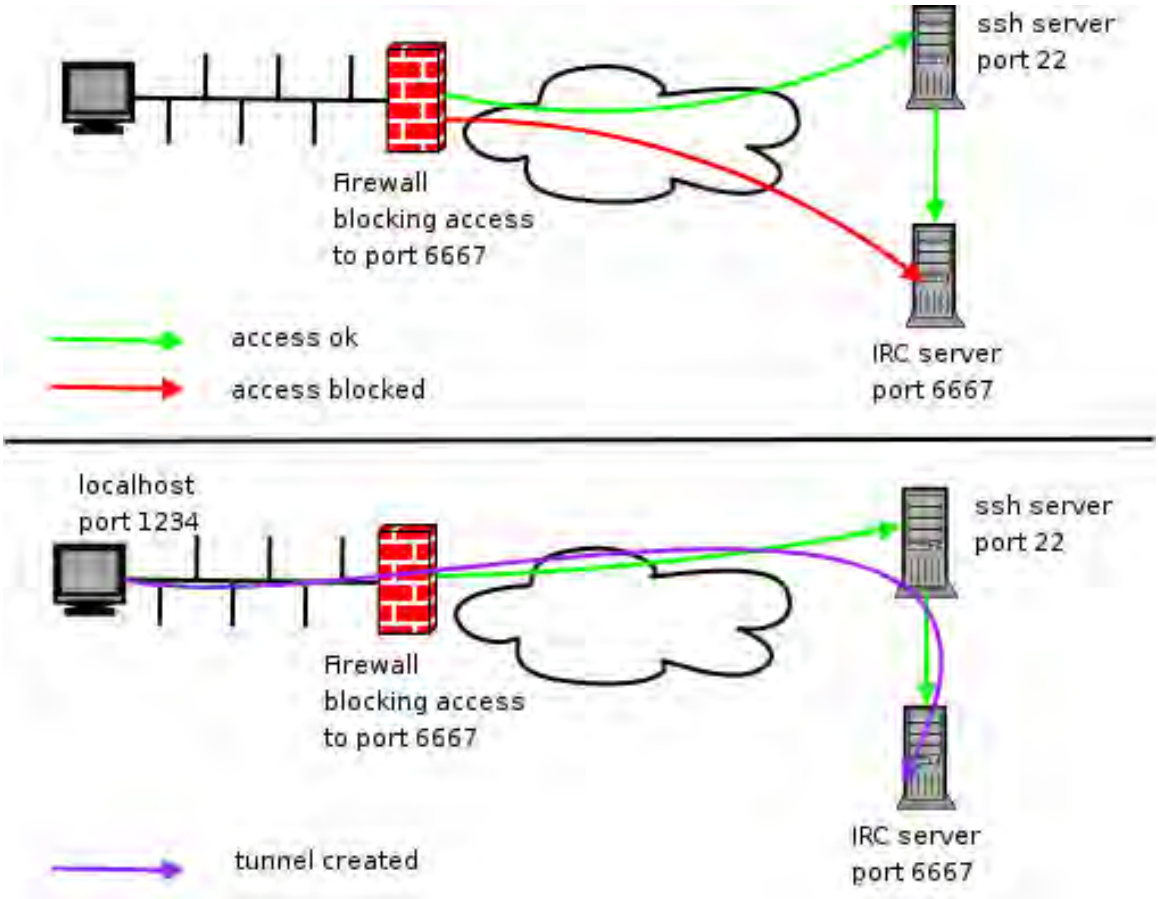

Figure 16 Example of port forwarding across a remote secure shell server.

Port forwarding[49] or port mapping, as it is sometimes referred, is the act of forwarding a network port from one network node to another, virtually creating a path across the network, in the case at hand, from the Internet side that is connected to the router or firewall to a computer inside the LAN permuting an external user to reach a port on a private IP address (inside a LAN) from the outside. Otherwise the computer inside the LAN couldn't be accessible and P2P wouldn't be able to fully work (it could contact outside machines but it wouldn't be fully visible from the outside). Due to the great variety of different implementations of GUIs that ultimately permits the same functionality, making the needed changes isn't a trivial process and one solution will not fit all instances.

The first fact that one should be aware of is that routers or the computer that is connected to the Internet will have a distinct IP addresses, one is provided by your ISP and is visible from the Internet the other will be visible only inside the LAN, this one can be assigned by the user or will default to a factory or Operative System factory setting.

To perform any changes to the configuration of the router you will need to know the IP that you can access its configuration page and login into it, if using a computer this same task can be done locally or remotely if the software allows it. Some minimum knowledge

49 `http://en.wikipedia.org/wiki/Port%20forwarding`

on how IPs and ports, even protocols (some configurations permit to distinguish between TCP and UDP packets) are used is required from the user to perform those tasks, as such this can become complicated to the normal user.

Note:
For security reasons don't leave the default passwords unchanged, as this will allow outsiders to control your property or leach your bandwidth.

3.5 Unique ID

Numbers control your life. Over time anyone identified by multitude of numbers. Like a phone number, your credit card numbers, the driver's license number, the social security number, even zip codes or your car license plate. They way this unique numbers are created, attributed and verified is fascinating. The Unique ID site (`http://www.highprogrammer.com/alan/numbers/`) by Alan De Smet provides information on the general subject. To our particular subject what is relevant is the algorithms behind it all. How to establish a unique identifiers for users and resources.

3.5.1 Universally Unique Identifiers (UUID) / Globally Unique Identifier (GUID)

For a P2P protocol/application to be able to manage user identification, authentication, build a routing protocol, identify resources etc... there is a need for (a set of) Unique Identifiers. While a true peer to peer protocol doesn't intend to establish a centralized service, it can frequently make use of an already established such service. So for instance, Usenet makes use of domain names to create globally unique identifier for articles. If the protocol refrains from even make use of such a service then uniqueness can only be based on random numbers and mathematical probabilities.

The problem of generating unique IDs can be broken down as uniqueness over space and uniqueness over time which, when combined, aim to produce a globally unique sequence. This leads to a problem detected over some P2P networks using Open Protocols/Multiple *vendors* implementations, due to the use of different algorithms on the generation of the **GUIDs** the uniqueness over space is broken leading to sporadic collisions.

UUIDs are officially and specifically defined as part of the ISO-11578 standard other specifications also exist, like RFC 4122[50], ITU-T Rec. X.667[51].

Examples of uses

1. Usenet article IDs.
2. In Microsoft's Component Object Model (COM) morass, an object oriented programming model that incorporates MFC (Microsoft Foundation Classes), OLE (Object Linking Embedding), ActiveX, ActiveMovie and everything else Microsoft is hawking

50 `http://www.ietf.org/rfc/rfc4122.txt`
51 `http://www.itu.int/ITU-T/studygroups/com17/oid.html`

lately, a GUID is a 16 byte or 128 bit number used to uniquely identify objects, data formats, everything.
3. The identifiers in the windows registry.
4. The identifiers used in used in RPC (remote procedure calls).
5. Within ActiveMovie, there are GUID's for video formats, corresponding to the FOURCC's or Four Character Codes used in Video for Windows. These are specified in the file uuids.h in the Active Movie Software Developer Kit (SDK). ActiveMovie needs to pass around GUID's that correspond to the FOURCC for the video in an AVI file.

Security There is a know fragility on UUIDs of version 1 (time and node based), as they broadcast the node's ID.

Software implementation

Programmers needing to implement UUID could take a look on these examples:

- OSSP uuid (`http://www.ossp.org/pkg/lib/uuid/`) is an API for ISO C, ISO C++, Perl and PHP and a corresponding CLI for the generation of DCE 1.1, ISO/IEC 11578:1996, and RFC4122 compliant Universally Unique Identifiers (UUIDs). It supports DCE 1.1 variant UUIDs of version 1 (time and node based), version 3 (name based, MD5), version 4 (random number based), and version 5 (name based, SHA-1). UUIDs are 128-bit numbers that are intended to have a high likelihood of uniqueness over space and time and are computationally difficult to guess. They are globally unique identifiers that can be locally generated without contacting a global registration authority. It is Open Sourced under the MIT/X Consortium License.

3.6 Hashes, Cryptography and Compression

Most P2P systems have to implement at least one algorithms for Hashing, D/Encryption and De/Compression, this section will try to provide some ideas of this actions in relation to the P2P subject as we will address this issues later in other sections.

Detailed information on the subject can be found on the Cryptography[52] Wikibook (`http://en.wikibooks.org/wiki/Cryptography`).

One way of creating structured P2P networks is by maintaining a Distributed Hash Table (DHT), that will server as a distributed index of the resources on the network.

Another need for cryptography is in the protection of the integrity of the distributed resources themselves, to make them able to survive an attack most implementations of P2P some kind of Hash function (MD5, SHA1) and may even implement a Hash tree designed to detect corruption of the resource content as a hole or of the parts a user gets (for instance using Tiger Tree Hash).

52 `http://en.wikibooks.org/wiki/Cryptography`

3.6.1 Hash function

A **hash function**[53] is a reproducible method of turning some kind of data[54] into a (relatively) small number that may serve as a digital "fingerprint" of the data. The algorithm[55] substitutes or transposes the data to create such fingerprints. The fingerprints are called **hash sums**, **hash values**, **hash codes** or simply **hashes**. When you referring to hash or hashes some attention must be given since it can also mean the hash functions.

A group of hashes (the result of applying an hash function to data) will sometimes be referred as a bucked[56] or more properly a hash bucket. Most hash buckets if generated by a non colliding hash function will generate distinct hashes for a give data input, there are other characteristics that should be considered when selecting an hash function but it goes beyond the scope of this book, just remember to check if the hash function/algorithm you are implementing satisfies your requirements. Wikibooks has several books that covers hashes in some way, you can check more about the subject in Algorithm implementation[57] or Cryptography[58]

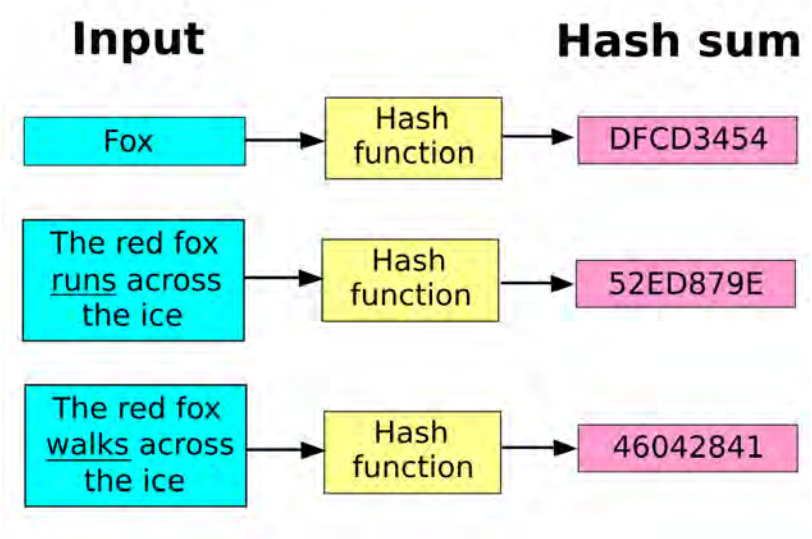

Figure 17 A typical hash function at work

Hash sums are commonly used as indices into hash table[59]s or hash file[60]s. Cryptographic hash function[61]s are used for various purposes in information security[62] applications.

Choosing a good hash function

53 http://en.wikipedia.org/wiki/Hash%20function
54 http://en.wikipedia.org/wiki/data
55 http://en.wikipedia.org/wiki/algorithm
56 http://en.wikipedia.org/wiki/Bucket%20%28computing%29
57 http://en.wikibooks.org/wiki/Algorithm%20implementation%2FHashing
58 http://en.wikibooks.org/wiki/Cryptography%2FHashes
59 http://en.wikipedia.org/wiki/hash%20table
60 http://en.wikipedia.org/wiki/hash%20file
61 http://en.wikipedia.org/wiki/Cryptographic%20hash%20function
62 http://en.wikipedia.org/wiki/information%20security

A good hash function is essential for good hash table performance. A poor choice of a hash function is likely to lead to clustering, in which probability of keys mapping to the same hash bucket (i.e. a collision) is significantly greater than would be expected from a random function. A nonzero collision probability is inevitable in any hash implementation, but usually the number of operations required to resolve a collision scales linearly with the number of keys mapping to the same bucket, so excess collisions will degrade performance significantly. In addition, some hash functions are computationally expensive, so the amount of time (and, in some cases, memory) taken to compute the hash may be burdensome.

One of the first hash algorithms used to verify the integrity of files in p2p systems (and in file transfers in general) was the MD5[63] created in 1992 (see rfc1321[64] **The MD5 Message-Digest Algorithm**). But as all hash most algorithms after some time some weaknesses where found, this sequence was repeated with the SHA1 algorithm and there is a high probability that others will fallow. Selecting the right tool for the job isn't enough, the programmer must continuously examine how the security of his choice is holding on in regards to the requirements placed on the selected hash algorithm.

Simplicity and speed are readily measured objectively (by number of lines of code and CPU benchmarks, for example), but strength is a more slippery concept. Obviously, a cryptographic hash function[65] such as SHA-1[66] (see Secure Hash Standard FIPS 180-1[67]) would satisfy the relatively lax strength requirements needed for hash tables, but their slowness and complexity makes them unappealing. However, using cryptographic hash functions can protect against collision attacks when the hash table modulus and its factors can be kept secret from the attacker, or alternatively, by applying a secret salt[68]. However, for these specialized cases, a universal hash function[69] can be used instead of one static hash.

In the absence of a standard measure for hash function strength, the current state of the art is to employ a battery of statistical[70] tests to measure whether the hash function can be readily distinguished from a random function. Arguably the most important test is to determine whether the hash function displays the avalanche effect[71], which essentially states that any single-bit change in the input key should affect on average half the bits in the output. Bret Mulvey advocates testing the *strict avalanche condition* in particular, which states that, for any single-bit change, each of the output bits should change with probability one-half, independent of the other bits in the key. Purely additive hash functions such as CRC[72] fail this stronger condition miserably.

63 http://en.wikibooks.org/wiki/Cryptography%2FMD5
64 http://tools.ietf.org/html/rfc1321
65 http://en.wikipedia.org/wiki/cryptographic%20hash%20function
66 http://en.wikipedia.org/wiki/SHA%20hash%20functions
67 http://www.itl.nist.gov/fipspubs/fip180-1.htm
68 http://en.wikipedia.org/wiki/salt%20%28cryptography%29
69 http://en.wikipedia.org/wiki/universal%20hash%20function
70 http://en.wikipedia.org/wiki/Statistics
71 http://en.wikipedia.org/wiki/avalanche%20effect
72 http://en.wikipedia.org/wiki/Cyclic%20redundancy%20check

Note:
CRC is often used to denote either the function or the function's output. A CRC can be used in the same way as a checksum[a] to detect accidental alteration of data during transmission or storage. CRCs are popular because they are simple to implement in binary hardware[b], are easy to analyze mathematically, and are particularly good at detecting common errors caused by noise in transmission channels. Historically CRCs have been given an ample used as an error detection/correction in telecommunications.

a http://en.wikipedia.org/wiki/checksum
b http://en.wikipedia.org/wiki/Computer%20hardware

For additional information on Hashing:

- Hashing Function Lounge[73] it includes a summary of what algorithms have known security flaws.

Collision avoidance

Implementing a Hash algorithm

Most hash algorithms[74] have an high degree of complexity and are designed for a specific target (hashing function) that may not apply, with the same level of guarantees, in other tasks. These algorithms (or raw descriptions of tem) are freely accessible, you can implement your own version or select to use an already existing and tested implementation (with a note for security concerns). Some examples of publicly available implementations are available at the Cryptography[75] Wikibook.

Note:
Keep in mind that some Hash functions may be subject to export restrictions.

There is a need for consistent results and repeatability, as you select and implement your hash algorithm, remember to run it over a batch of test vectors. If you are using a test framework[76] this should be added to your tests.

Hash tree (Merkle trees)

In cryptography[77], **hash trees** (also known as **Merkle trees**, ***invented in 1979 by Ralph Merkle[78]) are an extension of the simpler concept of hash list[79], which in turn is an extension of the old concept of hash[80]ing. It is a hash construct that***

73 http://paginas.terra.com.br/informatica/paulobarreto/hflounge.html
74 http://en.wikibooks.org/wiki/Algorithm%20Implementation%2FHashing
75 http://en.wikibooks.org/wiki/Cryptography%2FOpen%20Source%20Cryptography
76 http://en.wikipedia.org/wiki/List%20of%20unit%20testing%20frameworks
77 http://en.wikipedia.org/wiki/cryptography
78 http://en.wikipedia.org/wiki/Ralph%20Merkle
79 http://en.wikipedia.org/wiki/hash%20list
80 http://en.wikipedia.org/wiki/hash%20function

exhibits desirable properties for verifying the integrity of files and file subranges in an incremental or out-of-order fashion.

Hash trees where the underlying hash function is Tiger[81] (`http://www.cs.technion.ac.il/~biham/Reports/Tiger/`) are often called **Tiger trees** or **Tiger tree hashes**.

The main use of hash trees is to make sure that data blocks received from other peers in a peer-to-peer network[82] are received undamaged and unaltered, and even to check that the other peers do not send adulterated blocks of data. This will optimize the use of the Network and permit to quickly exclude adulterated content in place of waiting for the download of the hole file to complete to check with a single hash, an partial or complete hash tree can be downloaded and the integrity of each branch can be checked immediately (since they consist in "hashed" blocks or leaves of the Hash tree), even though the whole tree/content is not available yet, making also possible for the downloading peer to upload blocks of an unfinished files.

Usually, a cryptographic hash function[83] such as SHA-1[84], Whirlpool[85], or Tiger[86] is used for the hashing. If the hash tree only needs to protect against unintentional damage, the much less secure checksum[87]s such as CRCs[88] can be used.

In the top of a hash tree there is a *top hash* (or *root hash* or *master hash*). Before downloading a file on a P2P network, in most cases the top hash is acquired from a trusted source (a Peer or a central server that has elevated trust ratio). When the top hash is available, the hash tree can then be received from any source. The received hash tree is then checked against the trusted top hash, and if the hash tree is damaged or corrupted, another hash tree from another source will be tried until the program finds one that matches the top hash.

This requires several considerations:

1. What is a trusted source for the root hash.
2. A consistent implementation of the hashing algorithm (for example the size of the blocks to be transfered must be known and constant on every file transfer).

Tiger Tree Hash (TTH)

The Tiger tree hash is one of most useful form of hash tree on P2P Networks. Based on the cryptographic hash Tiger[89] created in 1995 by Eli Biham and Ross Anderson (see `http://www.cs.technion.ac.il/~biham/Reports/Tiger/` for the creator's info and C source example). The hash algorithm was designed with modern CPU in mind, in particular when dealing with working in 64-bit, it is one of the fastest and secure hashes on 32-bit machines.

81 `http://en.wikipedia.org/wiki/Tiger%20%28hash%29`
82 `http://en.wikipedia.org/wiki/Peer-to-peer`
83 `http://en.wikipedia.org/wiki/cryptographic%20hash%20function`
84 `http://en.wikipedia.org/wiki/SHA%20hash%20functions`
85 `http://en.wikipedia.org/wiki/Whirlpool%20%28hash%29`
86 `http://en.wikipedia.org/wiki/Tiger%20%28hash%29`
87 `http://en.wikipedia.org/wiki/checksum`
88 `http://en.wikipedia.org/wiki/Cyclic%20redundancy%20check`
89 `http://en.wikipedia.org/wiki/Tiger%20%28cryptography%29`

The TTH uses a form of binary hash tree, usually has a data block size of 1024-byte[90]s and uses the cryptographically secure Tiger hash[91].

Tiger hash is used because it's fast (and the tree requires the computations of a lot of hashes), with recent implementations and architectures, TTH is as fast as SHA1, with more optimization and the use of 64-bit processors, it will become faster, even though it generates larger hash values (192 bits vs. 160 for SHA1).

Tiger tree hashes are used in the Gnutella[92], Gnutella2[93], and Direct Connect[94] and many other P2P[95] file sharing protocols and in file sharing[96] in general.

A step by step introduction to the uses of the TTH was available as part of the Tree Hash Exchange (THEX) format (see page at the WEB Archive[97]).

Hash table

In computer science[98], a **hash table**[99], or a **hash map**, is a data structure[100] that associates *keys* with *values*. The primary operation it supports efficiently is a *lookup*: given a key, find the corresponding value. It works by transforming the key using a hash function[101] into a *hash*, a number that is used to index into an array to locate the desired location ("bucket") where the values should be.

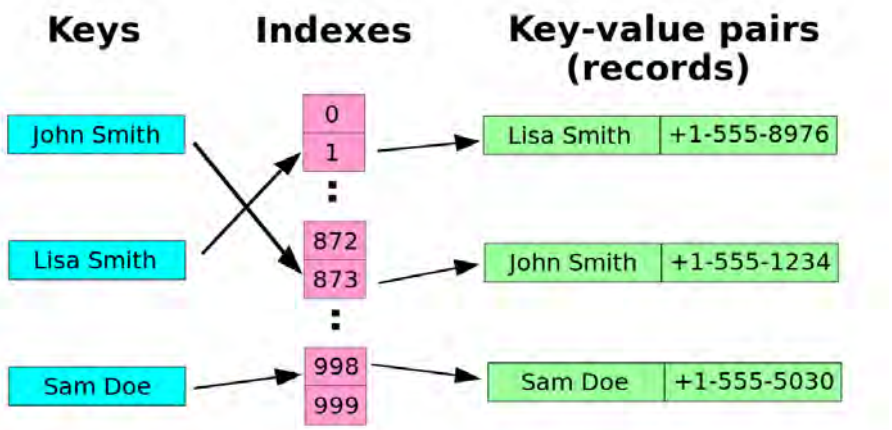

Figure 18 A small phone book as a hash table.

90 http://en.wikipedia.org/wiki/byte
91 http://en.wikipedia.org/wiki/Tiger%20%28hash%29
92 http://en.wikipedia.org/wiki/Gnutella
93 http://en.wikipedia.org/wiki/Gnutella2
94 http://en.wikipedia.org/wiki/Direct%20Connect%20%28file%20sharing%29
95 http://en.wikipedia.org/wiki/Peer-to-peer
96 http://en.wikipedia.org/wiki/file%20sharing
97 http://web.archive.org/web/20060409093503/www.open-content.net/specs/draft-jchapweske-thex-02.html
98 http://en.wikipedia.org/wiki/computer%20science
99 http://en.wikipedia.org/wiki/Hash%20table
100 http://en.wikipedia.org/wiki/data%20structure
101 http://en.wikipedia.org/wiki/hash%20function

Hash tables support the efficient addition of new entries, and the time spent searching for the required data is independent of the number of items stored (i.e. O(1)[102].)

In P2P system Hash tables are used locally on every client/server application to perform the routing of data or the local indexing of files, this concept is taken further as we try to use the same system in a distributed way, in that case distributed hash tables are used to solve the problem.

Distributed Hash Table (DHT)
The Distributed hash tables (DHTs)[103] concept was made public in 2001 but very few did publicly-release robust implementations.

Protocols

- **Content addressable network (CAN)[104]**
- **Chord[105]** (`http://pdos.csail.mit.edu/chord/`) - aims to build scalable, robust distributed systems using peer-to-peer ideas. It is completely decentralized and symmetric, and can find data using only log(N) messages, where N is the number of nodes in the system. Chord's lookup mechanism is provably robust in the face of frequent node failures and re-joins. A single research implementation is available in C but there are other implementations in C++, Java and Python.
- Tulip
- **Tapestry[106]**
- **Pastry[107]**
  - **Bamboo** (`http://bamboo-dht.org/`) - based on Pastry, a re-engineering of the Pastry protocols written in Java and licensed under the BSD license.

3.6.2 Encryption

A part of the security of any P2P Network, encryption is needed to make make sure only the "allowed" parties have access to sensitive data. Examples are the encryption of the data on a server/client setup (even on P2P) were clients could share data without fear of it being accessed on the server (a mix of this is if the Network would in itself enable a distributed cache mechanism for transfers, server-less), encryption of transfers, to prevent man-in-the-middle attacks, or monitor of data (see FreeNet[108]) and many other applications with the intent of protecting the privacy and enable an extended level of security to Networks.

There are several algorithm that can be used to implement encryption most used by P2P project include: BlowFish[109].

102 `http://en.wikipedia.org/wiki/Big-O%20notation`
103 `http://en.wikipedia.org/wiki/Distributed%20hash%20table`
104 `http://en.wikipedia.org/wiki/Content%20addressable%20network`
105 `http://en.wikipedia.org/wiki/Chord%20%28DHT%29`
106 `http://en.wikipedia.org/wiki/Tapestry%20%28DHT%29`
107 `http://en.wikipedia.org/wiki/Pastry%20%28DHT%29`
108 `http://en.wikipedia.org/wiki/FreeNet`
109 `http://en.wikipedia.org/wiki/Blowfish%20%28cipher%29`

3.6.3 Compression

3.7 Resources (Content, other)

P2P applications can be used to share any type of digital assets in the form of packed information that can be uniquely identified. P2P applications are well know for sharing files, this raises several issues and possibilities.

3.7.1 Metadata

Metadata[110] (meta-data, or sometimes meta-information) is "data about data". An item of metadata may describe an individual datum[111], or content item, or a collection of data including multiple content items and hierarchical levels, for example a database schema.

Data[112] is the lowest level of abstraction for knowledge, information is the next, and finally, we have knowledge highest level among all three. Metadata consists on direct and indirect data that help define the knowledge about the target item.

The hierarchy of metadata descriptions can go on forever, but usually context or semantic understanding makes extensively detailed explanations unnecessary.

The role played by any particular datum[113] depends on the context. For example, when considering the geography of London, "E83BJ" would be a datum and "Post Code" would be metadatum. But, when considering the data management of an automated system that manages geographical data, "Post Code" might be a datum and then "data item name" and "5 characters, starting with A – Z" would be metadata.

In any particular context, metadata characterizes the data it describes, not the entity described by that data. So, in relation to "E83BJ", the datum "is in London" is a further description of the place in the real world which has the post code "E83BJ", not of the code itself. Therefore, although it is providing information connected to "E83BJ" (telling us that this is the post code of a place in London), this would not normally be considered metadata, as it is describing "E83BJ" *qua* place in the real world and not *qua* data.

Difference between data and metadata

Usually it is not possible to distinguish between (plain) data and metadata because:

- Something can be data and metadata at the same time. The headline of an article is both its title (metadata) and part of its text (data).
- Data and metadata can change their roles. A poem, as such, would be regarded as data, but if there were a song that used it as lyrics, the whole poem could be attached to an audio file of the song as metadata. Thus, the labeling depends on the point of view.

These considerations apply no matter which of the above definitions is considered, except where explicit markup is used to denote what is data and what is metadata.

110 http://en.wikipedia.org/wiki/Metadata
111 http://en.wikipedia.org/wiki/datum
112 http://en.wikipedia.org/wiki/Data
113 http://en.wikipedia.org/wiki/datum

Hierarchies of metadata

When structured into a hierarchical arrangement, metadata is more properly called an ontology[114] or schema[115]. Both terms describe "what exists" for some purpose or to enable some action. For instance, the arrangement of subject headings in a library catalog serves not only as a guide to finding books on a particular subject in the stacks, but also as a guide to what subjects "exist" in the library's own ontology and how more specialized topics are related to or derived from the more general subject headings.

Metadata is frequently stored in a central location and used to help organizations standardize their data. This information is typically stored in a Metadata registry[116].

Schema examples

A good example on work being done on Metadata and how to make it accessible and useful can be examined in the W3C[117]'s old page Metadata Activity Statement (`http://www.w3.org/Metadata/Activity.html`) and the Resource Description Framework (RDF), that by the use of a declarative language, provides a standard way for using XML to represent metadata in the form of statements about properties and relationships of items on the Web. The RDF has it's own and more up to data page at `http://www.w3.org/RDF/`.

Metadata registries

Free Services

- **MusicBrainz[118]** (`http://musicbrainz.org/`) is a community music metadatabase (nonprofit service) that attempts to create a comprehensive music information site. You can use the MusicBrainz data either by browsing this web site, or you can access the data from a client program — for example, a CD player program can use MusicBrainz to identify CDs and provide information about the CD, about the artist or about related information. MusicBrainz is also supporting MusicIP's Open FingerprintTM Architecture, which identifies the sounds in an audio file, regardless of variations in the digital-file details. The community provides a REST styled XML based Web Service.

3.7.2 Database & Index

3.7.3 Files

Users of a P2P application may select files as the resource to be shared across the network. The simplest and secure method is to just share the content of files in a immutable way, most P2P applications cover this need, in any case other simultaneous models can be build around the file resource and the number of possible state of those files, being the most complex the implementation of a distributed filesystem.

114 `http://en.wikipedia.org/wiki/Ontology%20%28computer%20science%29`
115 `http://en.wikipedia.org/wiki/schema`
116 `http://en.wikipedia.org/wiki/metadata%20registry`
117 `http://en.wikipedia.org/wiki/World%20Wide%20Web%20Consortium`
118 `http://en.wikipedia.org/wiki/MusicBrainz`

Sharing Files

Sharing Files on a P2P network consists in managing (indexing, enable searches and transfer) two distinct resources, the local files and the remote files.

Local Files

If the a P2P application has support for file sharing. The most common method for sharing of files is to enable the user to select one or more directories of the local file system volumes. This includes providing, in the application preferences, the capability to select the path(s) to the files to be shared and to where to put the downloaded files.

It is also common practice to automatically add the download directory to the list of shared paths, or the downloaded files, so that the peer will immediately contribute to the replication of the downloaded resources.

Note:
It should be a concern to the implementor to stress to the user the importance about security and legal matters regarding the sharing of files on the network[a].

a `http://en.wikipedia.org/wiki/File%20sharing`

Monitor for Changes

After having the resources determined by the user, it is up to the application to as it is running verify any deletion, renaming or write action in general to any of the shared files since this could constitute a change to the file status (content change) or any addition of files, or alterations to the shared resources in general. This would could invalidate previously generated hashes or indexes. This problem can be addressed by generating an hash for the file content and use the OS control over the file-system to monitor changes as they happen (for instance on Windows one could use the Win32 API ReadDirectoryChangesW).

Depending on the requirements of the implementation, it may be beneficial to use multiple content hashes, a stronger to be used across the network and s faster and lower quality just to monitor local changes.

Detected shared files and changes to them

Since the application is not expected to be running all the time, and able to consistently monitor any changes done to the resources one must take steps to maintain further integrity. Because if the application is closed it will have no way to continue to monitor for changes and it will be up to the OS to permit some kind of API to detect file changes, like the last write access timestamp. So the steps needed to guarantee integrity each time the application is run is:

1. Verify that the resources (know files and shared paths) remain valid (they still exist and are unaltered).
2. Update the local index to reflect what was found in the above step (remove any invalid directories or volumes from the shared list). One can even prompt the user for corrective actions, as some my be removable media (DVDs etc...)
3. Start monitoring new changes.

4. Check all the resources content for changes since the last time the application exited, this may mean that each file must be checked individually for the last write date (most modern OS permit this). Depending on how it is implemented the next to steps can be included so to reduce duplication of work.
5. Verify that all previous indexed files are still present.
6. Verify for the existence of new files.

Local paths of significance

There are at least two paths with an higher degree of usefulness for the P2P application. The path were the application resides, since it can be needed for updating or for any required security checks and even possibly to serve as the base to locate the application preferences (if not using another setup, like the system registry on the Windows OS) and/or the download directory, a write enabled directory, where all downloads will be put.

Note:
The download directory should be monitored for the available free space, and if the content is automatically shared it should be a security concern as it will be prime vector for attacks, more than any other shared paths were a usurpation from an external source could not only make unwanted content available to the outside but enable external entities to upload into the local machine dangerous content.

External Files
This are the files that aren't directly under the local user control.

File Filtering
Most P2P application do not enable selecting single files for sharing but do it in bulk by enabling the selection of specific directories, there is a need to provide a mechanism to exclude specific files from being shared, by the will of the user, the specific of the network or even to reduce the pollution or the filesharing system burden in handling unwanted files. Be it by size, extension or even the intrinsic content of the file having the ability to weed them out is useful.

3.7.4 Indexing

Indexing, the task of indexing resources, like shared file, has the objective of enabling the application to know what resources are available to be shared on the network.

Indexing occurs each time the application is run, since the resources can only be monitored for changes while the application runs and when a change to the shared resources preferences is performed. This also makes it a necessary to have during the application run-time a function that monitors changes to resources.

Most P2P application also support the exclusion or filtering of shared resources, for instance if a directory is selected to be shared but some of the content needs to be excluded.

Searches

Using a DHT

Metalink

3.8 Services

3.8.1 Seeds

3.8.2 Leechers

In Nature, cooperation is widespread but so too are leechers (cheats, mutants). In evolutionary terms, cheats should indeed prosper, since they don´t contribute to the collective good but simply reap the benefits of others' cooperative efforts, but they don't. Both compete for the same goal using different strategies, cooperation is the path of less cost to all (even for leechers on the long run), cooperation provides stability and previsibility and on the other hand if cheats are not kept in some sort of equilibrium they generate a degradation of the system that can lead to its global failure.

In computer science and especially on the Internet, being a leech or leecher refers to the practice of benefiting, usually deliberately, from others' information or effort but not offering anything in return, or only token offerings in an attempt to avoid being called a leech. They are universally derided.

The name derives from the leech, an animal which sucks blood and then tries to leave unnoticed. Other terms are used, such as freeloader, but leech is the most common.

Examples

- On peer to peer networks, a leecher shares nothing (or very little of little worth) for upload. Many applications have options for dealing with leeches, such as uploading at reduced rates to those who share nothing, or simply not allowing uploads to them at all. Some file-sharing forums have an anti-leech policy to protect the download content, where it will require users to expend more energy or patience than most leechers are willing to before they can access the "download area".
- Most BitTorrent sites refer to leeches as clients who are downloading a file, but can't seed it because they don't have a complete copy of it. They are by default configured to allow a certain client to download more when they upload more.
- When on a shared network (Such as a school or office LAN), any deliberate overuse of bandwidth (To the point at which normal use of the network would be noticeably degraded) can be called leeching.
- In online computer games (especially role-playing games), leeching refers to the practice of a player joining a group for the explicit purpose of gaining rewards without contributing anything to the efforts necessary to acquire those rewards. Sometimes this is allowed in an effort to power-level a player. Usually it is considered poor behavior to do this

without permission from the group. In first person shooters the term refers to a person who benefits by having his team mates carry him to a win.

- Direct linking is a form of bandwidth leeching that occurs when placing an unauthorized linked object, often an image, from one site in a web page belonging to a second site (the leech). This constitutes an unauthorized use of the host site's bandwidth and content.

In some cases, leeching is used synonymously with freeloading rather than being restricted to computer contexts.

Possible solutions

Trust & Reputation
Due the volatility and modularity of peer to peer networks, peer trust (mentioned in the reference to building communities) and user participation is one of the few motivations or lubricant for online transactions of services or resources that falls ultimately to the creator. Having an average good reputation in participants will also boost the trust on the network and the systems built upon it.

3.8.3 File sharing

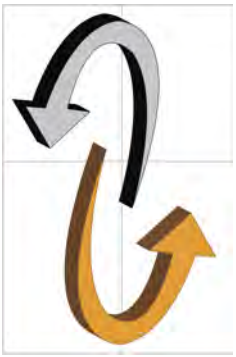

Figure 19
Sharing.

Traditionally, file transfers involve two computers, often designated as a client and a server and most operations are for the copying files from one machine to another.

Most WEB and FTP servers are punished for being popular. Since all uploading is done from one central place, a popular site needs more resources (CPU and bandwidth) to be able to cope. With the use of P2P, the clients automatically mirror the files they download, easing the publisher's burden.

One limitation of most P2P protocols is that they don't provide a complex file-system emulation or a user-right system (permissions), so complex file operations like NFS or FTP protocols provide are very rare, this also has a reason to be, since the networks is decentralized a system for the authentication of users is hard to implement and most are easy to break.

Another concept about P2P transfers is the use of bandwidth, things will not be linear, transfers will depend on the availability of the resource, the load of the seeding peers, size of the network and the local user connection and load on its bandwidth.

As seen before, Downloading files with a restrictive copyright, license or under a given country law, may increase the risk of being sued. Some of the files available on these networks may be copyrighted or protected under the law. You must be aware that there is a risk involved.

Receiving

from multiple sources (segmented downloading, swarm)

Multiple source download, (segmented download, swarming download), can be a more efficient way of downloading[119] files[120] from many peers at once. The one single file is downloaded, in parallel, from several distinct sources or uploaders of the file. This can help a group of users with asymmetric[121] connections, such as ADSL[122] to provide a high total bandwidth[123] to one downloader, and to handle peaks in download demand. All swarm transfers depends on at least the existence of one full complete copy of the wanted file (a possible evolution could include adding some sort of Reed–Solomon (RS) Algorithm[124] into the mix) and it is mostly useful for large files (size depending on the available bandwidth, since the trade off includes a higher cost in CPU and extraneous data transfers to make the system work.

This technique can not magically solve the problem, in a group of users that has insufficient upload-bandwidth, with demand[125] higher than supply[126]. It can however very nicely handle peaks, and it can also to some degree let uploaders upload "more often" to better utilize their connection. However, naive implementations can often result in file corruption, as there is no way of knowing if all sources are actually uploading segments of the same file. This has led to most programs using segmented downloading using some sort of checksum[127] or hash algorithm[128] to ensure file integrity.

119 http://en.wikipedia.org/wiki/downloading
120 http://en.wikipedia.org/wiki/file
121 http://en.wikipedia.org/wiki/asymmetric
122 http://en.wikipedia.org/wiki/ADSL
123 http://en.wikipedia.org/wiki/bandwidth
124 http://en.wikipedia.org/wiki/Reed%E2%80%93Solomon%20error%20correction
125 http://en.wikipedia.org/wiki/demand
126 http://en.wikipedia.org/wiki/supply
127 http://en.wikipedia.org/wiki/checksum
128 http://en.wikipedia.org/wiki/hash%20algorithm

Resuming

Preview while Downloading

Is may help to let users preview files before the downloading process finishes and as soon as possible, this will improve the quality of shares on the network increasing users confidence and reducing lost time and bandwidth.

Security

Poisioning and Pollution

Examples include:

- poisoning attacks (e.g. providing files whose contents are different from the description)
- polluting attacks (e.g. inserting "bad" chunks/packets into an otherwise valid file on the network)
- insertion of:
  - viruses to carried data (e.g. downloaded or carried files may be infected with viruses or other malware)
  - malware[129] in the peer-to-peer network software itself (e.g. distributed software may contain spyware)

Rights Protection

Protecting the copyrights over content shouldn't be imposed on the public, even if most copyrights policies today have removed the need to state that a work has reserved rights, there shouldn't be any expectation that the general public would have to go out of way to protect the benefits of a minority even more if they aren't clearly stated. It is expected that the works in the public domain will today outnumber works that have valid copyrights.

Given the actual situation, any p2p application will have the moral duty to inform users of the issues risen due to this very confusing situation. Some solutions to ease the problem and help copyright holders to protect their works have over time came forward but no interest seems to have been raised by those that should care about the issue. This section will try to show the possibilities available.

A simple solution to this issue would be to clearly put the responsibility of identifying work under copyright to the right owners, to implement this solution a database needs to be freely available to p2p applications so content can be declared as copyrighted or a standard must be created to clearly identify those works. In response to this problem several solution were discussed and proposed but in July 2003, a solution by the name of BluFilter was put forward by the Kokopelli Network Inc. (co-founded by Alex Sauriol), that would permit to identify the copyright status, by comparing the waveforms that are stored within the mp3 (see kokopellinetworks.com @ web.archive.org[130]), this approach seemed viable but no adoption of a similar technology seems to be openly used to inform users but rumors have appeared that similar approaches have been used as a base legal litigations.

129 http://en.wikipedia.org/wiki/malware
130 http://web.archive.org/web/20041130222253/kokopellinetworks.com/

3.8.4 Distributed Proxy

3.8.5 VoIP

3.8.6 Distributed Streaming

3.8.7 Priority settings

Enabling the dynamic set of priorities on transfers will not only keep users happy but provide an easy way to boost transfers on highly sicked content increasing the speed of replication of the same on the network.

One can even go a step further and permit a by resource configuration, enabling the removal or configuration access rights to each resource, like on a file system or even enable a way to permit a market for free trading of data letting users set a specific ratio for that resource.

3.8.8 Bandwidth Scheduler

Managing local resources is important not only to the local user but to the global network. Managing and enabling control of the application use of bandwidth will serve as incentive for users to improve how they manage that resource (what and how it is being used) and if taken in consideration by the application as a dynamic resource it can have positive effects on the global network by reducing wasting.

Many of the actual P2P applications enable users such control of their bandwidth, but not only P2P benefits from this strategy, today with a significant part of most computers connected to the Internet, managing this scarce resource is of top most importance. One example is Microsoft's Background Intelligent Transfer Service (BITS)[131] aimed at enabling system updates or even the MS IM service to transfer data whenever there is bandwidth which is not being used by other applications, of note is also the ability to use the BITS technology since it is exposed through Component Object Model[132] (COM).

3.8.9 "New" models

Fault-Tolerant Web Sites

Many people have speculated that peer-to-peer file sharing technology could be used to improve wiki and other kinds of Internet services.

High quality video or large files distribution

The Internet infrastructure was not designed to support broadcasting. P2P partially solves this infrastructural bottleneck by switching the server or content provider from a single

131 http://en.wikipedia.org/wiki/Background%20Intelligent%20Transfer%20Service
132 http://en.wikipedia.org/wiki/Component%20Object%20Model

point to a decentralized infrastructure, that depends not on the specific network limitations but on the protocol that optimizes the distribution and its popularity.

In February 2008 the European Union announced its commitment into a four-year project that aims to create an open source, peer-to-peer BitTorrent-like client called P2P-Next, based on an improvement of the Delft University of Technology python project Tribler. The EU will contribute 14 million euros (£10.5 million, $22 million) into this project and another 5 million euros (£3.7 million, $7.4 million) will be added by another 21 partners that includes the European Broadcasting Union, Lancaster University, BBC, Markenfilm, VTT Technical Research Center and Pioneer Digital Design Center Limited.

Video distribution

Even if several of the existing P2P applications do support video distribution, rarely have they had a consistent infrastructure and the main goal to permit subscribing to video content. Several do provide searches but a careful integration of a video player/media manager with the distribution is so far extremely rare and an evolving proposition.

Considering the door the centralized systems like YouTube, Google Video and their ilk opened in regards to online video, this new need for distribution has lead to the creation of several P2P offering, like Joost (from the same team as the Kazaa and Skype[133] applications), Tape it off the Internet or Veoh TV.

Of particular notice among the new emerging platforms is Miro (formerly Democracy Player), one of the first projects of this kind. Available at `http://www.getmiro.com/` , it is open-source (supports all current versions of Windows, Mac OS and several Linux distro) non-profit video player that supports practically every format. Created by the Participatory Culture Foundation (PCF, a non-profit organization), it automates subscribing to video RSS feeds that are then downloaded using Bittorrent that reduces the money spent on bandwidth required for distribution. Available is also the Miro Video Converter application, that promotes the video conversion to the patent-free OGG Theora codec[134] (which provides the same quality of the non free h.264) and has been backing the campaign to get Video on Wikipedia.

Real time video

Transmission of live events to millions of people using the actual infrastructure imposes limits on the quality of the output and high expectations on the hardware resources, not only on network resources but on the encoding and playback capabilities on each side of the transfer.

This has lead to the use of P2P network, as an attempt to save server bandwidth. An example of this new approach is the MSR Asia Peer-to-peer Video Broadcast System (`http://research.microsoft.com/en-us/projects/p2pbroadcast/`) from Microsoft Research Asia that was used in the 2008 Beijing Olympic Games claiming the use of more than two million Internet peers.

133 `http://en.wikibooks.org/wiki/The%20Computer%20Revolution%2FCommunication%2FSkype`

134 `http://en.wikibooks.org/wiki/FOSS%20Open%20Standards%2FComparison%20of%20File%20Formats%23Theora`

Distribute digital content over semi-private P2P networks

P2P has also been used in recent times to help ease the load on distribution digital content in general in a system of restricted participation (non-public access) and dedicated function.

BitTorrent DNA[135] or **BitTorrent Delivery Network Accelerator**, reportedly available since January 2005 operates in this fashion. This distribution technology has been especially interesting for games software houses, that have been moving their business to the Internet and adopting the business practice of subscription to services or content, in place of selling the ownership of a copy of the digital goods they create, or in a mix model. This will require a network connection to download extra content often protected by DRM, the number of transfers increases, in a way that it is not uncommon that during the update process or even the installation of the product the user's machine will be utilized to create a P2P network, sometimes even without the full knowledge of the consumer, to help the distribution of that paid content.

Blizzard[136] for instance, uses BitTorrent and does inform users that when downloading some contents the user will also be sharing it across a P2P network.

Solid State Networks already offers a P2P-based delivery solution as does Akamai[137], that sells Netsession Interface to game publishers. There are also other tools that are used to the same effect (not exclusive to games) but as helpers to the distribution of large files, Pando Media Booster is an example.

There is however a problems about transparency over the utilization of the costumers' resources. Solid State Networks recently, and after several costumers reported issues regarding this new approach, that often degrades users systems security, stability and performance. Has started a campaign to establish an industry "best practices", or most of this solutions will fall in the definition of malware, `http://www.solidstatenetworks.com/index.php/about-us/p2p-best-practices/` (PDF[138]).

P2P is also used for digital art, for instance the Electric sheep[139] project (`http://community.electricsheep.org`), a C++ open source based on libtorrent BitTorrent implementation, is a distributed computing[140] project for animating and evolving fractal flame[141]s, which are in turn distributed to the networked computers, which display them as a screensaver[142]. The process is transparent to the casual user, who can simply install the software as a screensaver. Alternatively, the user may become more involved with the project, manually creating sheep (video files of animated fractal flames) for upload to the server.

135 `http://en.wikipedia.org/wiki/BitTorrent%20DNA`
136 `http://en.wikipedia.org/wiki/Blizzard`
137 `http://en.wikipedia.org/wiki/Akamai`
138 `http://www.solidstatenetworks.com/bestpractices/SSN-Best-Practices.pdf`
139 `http://en.wikipedia.org/wiki/Electric%20Sheep`
140 `http://en.wikipedia.org/wiki/distributed%20computing`
141 `http://en.wikipedia.org/wiki/fractal%20flame`
142 `http://en.wikipedia.org/wiki/screensaver`

Hardware

As P2P protocols become more mature and claims to have a good level of public adoption, that we will see more embedded devices supporting the P2P solutions. This is already beginning to happen:

- TonidoPlug (`http://www.tonidoplug.com`) a tiny, low-power, low-cost personal home server and NAS device. Tonido Torrent is a web-based torrent client based on the BitTorrent implementation in the Boost licensed libtorrent library.
- Excito (`http://excito.com`) offers also a range of servers, named "Buba" that supports the BitTorrent protocol by using the the Boost licensed libtorrent library.

Traffic Shapers

Set Top Boxes

P2P technologies can also be used to provide a means to at low cost distribute content in an automated way.

Using a peer to peer architecture directly connected to a broadband line, a set top box (a stripped down PC of sorts), with an operating software and some storage space can for instance provide a service similar to video on demand.

VUDU

VUDU (`http://www.vudulabs.com/`), thousands of movies delivered directly to your TV, it doesn't require a PC and is independent of your cable or satellite TV service.

Distributed File-systems

Distributed File-systems aren't new but pre-P2P system depended on a server (or the election of a server from a pool of known machines) and were prominently focused on LANs that provided increased stability to the network. New systems are more reliable facing the volatility of a network and implement the new technologies P2P relies on. Most implementation of a P2P distributed File-systems will have evolved based on the FreeNet[143] model to some degree.

Tahoe Least-Authority Filesystem (Tahoe-LAFS)

Tahoe-LAFS (`http://allmydata.org/trac/tahoe/wiki)` a secure remote (distributed) filesystem, released under the GNU's General Public License (GPL), that shares with P2P the underlying network architecture and the principle of least authority, but it is not entirely decentralized. It requires a central node, called an Introducer, needed to connect new nodes.

With the objective of creating a fault-tolerant storage pool across several peers (cloud storage) were everybody provides storage for each other. The files are distributed across

143 `http://en.wikipedia.org/wiki/FreeNet`

the multiple nodes using AES encryption. A variation of Reed-Solomon error correction is used to permit peers to disconnect without affecting the integrity of the content.

Tahoe is free software: all of the source code is available under an open-source license. The home page is at `http://allmydata.org` . Tahoe is sponsored by allmydata.com, which uses it as the back-end for a commercial personal-data backup service.

Omemo (http://www.omemo.com/)
Omemo[144] (`http://www.omemo.com/`) is a free and open source (Visual Basic) P2P application under the GPL, from Pablo Soto[145] creator of the MANOLITO protocol and Blubster. Omemo takes a different approach and uses a ring-shaped DHT based on Chord. It is meant to support key based routing while keeping query source obscurity due to randomization. Available for Windows.

Mobile Peer-to-Peer Computing

PEPERS Project
The PEPERS project (`http://www.pepers.org/`) focus is to design, implement and validate a reliable platform with high-level support for the design, development and operational deployment of secure mobile peer-to-peer applications for future Ambient Intelligent (AmI) environments. The platform will greatly assist the work and benefit both mobile application and service providers and service users. The project will address issues related to security, privacy, trust and access control in mobile peer-to-peer (p2p) systems by proposing a relevant framework architecture. The framework will include support for policy-based security management for mobile systems. The specific thematic focus of the project (that reflects also in the selection of the pilot scenarios that will be implemented) is the collaboration among teams dispersed over a geographical area. The consortium partners come from 4 EU Member states (Greece, UK, Italy and Cyprus) and include key technology providers and industry players, and leading academic institutions, as well as users partners from diverse business domains (media and journalism, security services) with increased needs for secure collaboration through advanced technological solutions that bring significant expertise and knowledge in PEPERS related technologies as well as the underlying operating business models.

From a technical point of view the project will focus on:

- the definition of appropriate security services for mobile peer-to-peer applications over suitable protocols
- the analysis and design of possible platforms and interfaces in mobile devices that can provide security services to support peer-to-peer applications
- the definition of interfaces based on open standards that will allow secure mobile access to application servers

144 `http://en.wikipedia.org/wiki/Omemo`
145 `http://en.wikipedia.org/wiki/Pablo%20Soto`

The secure use of mobile peer-to-peer applications based on the project technology will be validated in real-life pilot applications supporting collaborative work in two operational domains:

- media and journalism: reporters working on assignment need to be able to record, edit and exchange information in a way that will protect and monitor intellectual property rights and the interests of the organization that employs them. This is particularly relevant as it covers both general issues of a mobile workforce and more specific issues related to the media application domain.
- physical security: guards and mobile patrols need to be able to receive and transmit sensitive customer information in a dynamic environment when they are called to respond and co-operate in case of ad-hoc exceptional situations.

Digital Currency

Bit Coin

Bitcoin (`http://www.bitcoin.org`), a peer-to-peer network based digital currency system without a central server or trusted parties, no central authority to issue new money or keep track of transactions. Nodes of the network hold the crypto keys to their own money and transact directly with each other, with the help of the network to check for double-spending.

The limited inflation of the Bitcoin system's money supply is distributed evenly (by CPU power) throughout the network.

- Money transfers through the Internet, without middlemen.
- No third parties control over transactions.
- Transactions are practically free.
- No inherent instability caused by fractional reserve banking or policies of central banks.

4 External Links

- Chord lookup service[1]
- PAST distributed storage utility[2]
- CoopNet cooperative content distribution system[3]
- More peer-to-peer research resources[4]
- Advanced Peer-Based Technology Business Models[5] - P2P Industry Model from M.I.T.
- FailSafeWiki[6]
- The Peer To Peer Foundation[7] site with discussion of the philosophy/economics behind peer-to-peer networks.
- Various Papers(Research & Practical) on P2P (GOBO)[8]
- P2P Networking 4 Science[9] A review of the current and potential uses of peer-to-peer networks in scientific research
- Annual IEEE International Conference on peer-to-Peer Computing[10]
- Conferences and Resources[11] on peer-to-peer computing
- IBM Developer Works: The practice of peer-to-peer computing[12]
- Internet2® peer-to-Peer Working Group[13]
- Microsoft peer-to-peer networking[14]
- OpenP2P[15] peer-to-peer development resources
- Introduction to Windows Peer-to-Peer Networking[16] from MSDN.
- Study: P2P effect on legal music sales "not statistically distinguishable from zero"[17]

[18]

1 http://www.pdos.lcs.mit.edu/chord/
2 http://www.research.microsoft.com/~antr/PAST/
3 http://research.microsoft.com/~padmanab/projects/coopnet/
4 http://www-2.cs.cmu.edu/~kunwadee/research/p2p/links.html
5 http://shumans.com
6 http://wikifeatures.wiki.taoriver.net/moin.cgi/FailSafeWiki
7 http://blog.p2pfoundation.net/
8 http://getonebyone.googlepages.com/researchpapers
9 http://www.firstauthor.org/Downloads/P2P.pdf
10 http://www.ida.liu.se/conferences/p2p/portal/
11 http://www.cs.ucd.ie/staff/aquigley/home/index.php?Links:P2P
12 http://www-106.ibm.com/developerworks/java/library/j-p2pcol.html
13 http://p2p.internet2.edu/
14 http://www.microsoft.com/p2p
15 http://www.openp2p.com/
16 http://www.microsoft.com/technet/prodtechnol/winxppro/deploy/p2pintro.mspx
17 http://arstechnica.com/news.ars/post/20070212-8813.html
18 http://en.wikibooks.org/wiki/Category%3A

5 Contributors

Edits	User
1	*nix[1]
1	ALargeElk[2]
15	Adrignola[3]
5	ArneBab[4]
1	Aya[5]
1	Bryan Derksen[6]
3	Chuckhoffmann[7]
4	Darklama[8]
1	DavidCary[9]
1	Derbeth[10]
2	Guanaco[11]
1	J36miles[12]
1	JeffyJeffyMan2004[13]
6	Jguk[14]
2	John Vandenberg[15]
1	LeeHunter[16]
2	Maximus Rex[17]
2	Merrheim[18]
1	Mike.lifeguard[19]
2	Mjchael[20]
3	Mshonle[21]

1 http://en.wikibooks.org/wiki/User:*nix
2 http://en.wikibooks.org/wiki/Special:Contributions/ALargeElk
3 http://en.wikibooks.org/wiki/User:Adrignola
4 http://en.wikibooks.org/wiki/Special:Contributions/ArneBab
5 http://en.wikibooks.org/wiki/User:Aya
6 http://en.wikibooks.org/wiki/Special:Contributions/Bryan_Derksen
7 http://en.wikibooks.org/wiki/User:Chuckhoffmann
8 http://en.wikibooks.org/wiki/User:Darklama
9 http://en.wikibooks.org/wiki/User:DavidCary
10 http://en.wikibooks.org/wiki/User:Derbeth
11 http://en.wikibooks.org/wiki/User:Guanaco
12 http://en.wikibooks.org/wiki/User:J36miles
13 http://en.wikibooks.org/wiki/Special:Contributions/JeffyJeffyMan2004
14 http://en.wikibooks.org/wiki/User:Jguk
15 http://en.wikibooks.org/wiki/User:John_Vandenberg
16 http://en.wikibooks.org/wiki/Special:Contributions/LeeHunter
17 http://en.wikibooks.org/wiki/User:Maximus_Rex
18 http://en.wikibooks.org/wiki/User:Merrheim
19 http://en.wikibooks.org/wiki/User:Mike.lifeguard
20 http://en.wikibooks.org/wiki/User:Mjchael
21 http://en.wikibooks.org/wiki/User:Mshonle

2 Mwtoews[22]
2 Newmanbe[23]
1444 Panic2k4[24]
1 PxT[25]
8 QUBot[26]
3 QuiteUnusual[27]
2 Siddord[28]
1 Sorrelvesper[29]
1 Sub[30]
2 Underscore[31]
1 Whiteknight[32]

22 http://en.wikibooks.org/wiki/User:Mwtoews
23 http://en.wikibooks.org/wiki/User:Newmanbe
24 http://en.wikibooks.org/wiki/User:Panic2k4
25 http://en.wikibooks.org/wiki/Special:Contributions/PxT
26 http://en.wikibooks.org/wiki/User:QUBot
27 http://en.wikibooks.org/wiki/User:QuiteUnusual
28 http://en.wikibooks.org/wiki/User:Siddord
29 http://en.wikibooks.org/wiki/User:Sorrelvesper
30 http://en.wikibooks.org/wiki/User:Sub
31 http://en.wikibooks.org/wiki/Special:Contributions/Underscore
32 http://en.wikibooks.org/wiki/User:Whiteknight

List of Figures

- GFDL: Gnu Free Documentation License. http://www.gnu.org/licenses/fdl.html
- cc-by-sa-3.0: Creative Commons Attribution ShareAlike 3.0 License. http://creativecommons.org/licenses/by-sa/3.0/
- cc-by-sa-2.5: Creative Commons Attribution ShareAlike 2.5 License. http://creativecommons.org/licenses/by-sa/2.5/
- cc-by-sa-2.0: Creative Commons Attribution ShareAlike 2.0 License. http://creativecommons.org/licenses/by-sa/2.0/
- cc-by-sa-1.0: Creative Commons Attribution ShareAlike 1.0 License. http://creativecommons.org/licenses/by-sa/1.0/
- cc-by-2.0: Creative Commons Attribution 2.0 License. http://creativecommons.org/licenses/by/2.0/
- cc-by-2.0: Creative Commons Attribution 2.0 License. http://creativecommons.org/licenses/by/2.0/deed.en
- cc-by-2.5: Creative Commons Attribution 2.5 License. http://creativecommons.org/licenses/by/2.5/deed.en
- cc-by-3.0: Creative Commons Attribution 3.0 License. http://creativecommons.org/licenses/by/3.0/deed.en
- GPL: GNU General Public License. http://www.gnu.org/licenses/gpl-2.0.txt
- LGPL: GNU Lesser General Public License. http://www.gnu.org/licenses/lgpl.html
- PD: This image is in the public domain.
- ATTR: The copyright holder of this file allows anyone to use it for any purpose, provided that the copyright holder is properly attributed. Redistribution, derivative work, commercial use, and all other use is permitted.
- EURO: This is the common (reverse) face of a euro coin. The copyright on the design of the common face of the euro coins belongs to the European Commission. Authorised is reproduction in a format without relief (drawings, paintings, films) provided they are not detrimental to the image of the euro.
- LFK: Lizenz Freie Kunst. http://artlibre.org/licence/lal/de
- CFR: Copyright free use.

- EPL: Eclipse Public License. `http://www.eclipse.org/org/documents/epl-v10.php`

Copies of the GPL, the LGPL as well as a GFDL are included in chapter Licenses[33]. Please note that images in the public domain do not require attribution. You may click on the image numbers in the following table to open the webpage of the images in your webbrower.

33 Chapter 6 on page 129

1	User:Mauro Bieg[34], User:Mauro Bieg[35]	
2	User:Mauro Bieg[36], User:Mauro Bieg[37]	LGPL
3	Caricature by J.J.[38], SVG file by Gustavb[39]	CC-BY-SA-3.0
4	Dyllan Horrocks (spelling based on signature in the image, may be mispelled)	
5	Iron Bishop[40], Iron Bishop[41]	GFDL
6	helix84[42] and (eMule original) eMule team[43], helix84[44] and (eMule original) eMule team[45]	GPL
7	User:K.lee[46], User:K.lee[47]	
8	Raúl Silva	
9	perfectska04[48]	GPL
10	Hilton William Ganzo Perantunes	
11	Bensin, Bilderbot, Fastily, Grafite, Kilom691, Maksim, Malyszkz, Waldir, YMS	
12	˜Pyb[49]	
13	/	
14	Luis F. Gonzalez	
15	Christoph Sommer	GFDL
16	This file is lacking **author** information.	
17	Bilderbot, Citypeek, Emijrpbot, Ggia, Hazard-Bot, Helix84, Joanjoc, LyingB, Mdd, Nguyễn Thanh Quang	
18	Emijrpbot, Hazard-Bot, Helix84, Simeon87, Xhienne, Yonidebest	
19	BotMultichill, Michael Barera, SchlurcherBot	

34 http:////commons.wikimedia.org/wiki/User:Mauro_Bieg
35 http:///wiki/User:Mauro_Bieg
36 http:////commons.wikimedia.org/wiki/User:Mauro_Bieg
37 http:///wiki/User:Mauro_Bieg
38 http:////en.wikipedia.org/wiki/User:J.J.
39 http:////en.wikipedia.org/wiki/User:Gustavb
40 http:////commons.wikimedia.org/wiki/User:Iron_Bishop
41 http:///wiki/User:Iron_Bishop
42 http:////commons.wikimedia.org/wiki/User:Helix84
43 http://www.emule-project.net/home/perl/general.cgi?l=1&rm=team
44 http:///wiki/User:Helix84
45 http://www.emule-project.net/home/perl/general.cgi?l=1&rm=team
46 http:////commons.wikimedia.org/wiki/User:K.lee
47 http:///wiki/User:K.lee
48 http://www.gnome-look.org/usermanager/search.php?username=perfectska04
49 http://flickr.com/photos/95753155@N00

6 Licenses

6.1 GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow. TERMS AND CONDITIONS 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion. 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work. 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary. 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures. 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee. 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

* a) The work must carry prominent notices stating that you modified it, and giving a relevant date. * b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices". * c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it. * d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate. 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

* a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange. * b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge. * c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b. * d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements. * e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying. 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

* a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or * b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or * c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or * d) Limiting the use for publicity purposes of names of licensors or authors of the material; or * e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or * f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way. 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10. 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so. 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it. 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law. 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy

both those terms and this License would be to refrain entirely from conveying the Program. 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such. 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version. 15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION. 16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. 17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author> This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'. This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

6.2 GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference. 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License. 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies. 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document. 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

* A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission. * B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement. * C. State on the Title page the name of the publisher of the Modified Version, as the publisher. * D. Preserve all the copyright notices of the Document. * E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices. * F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below. * G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice. * H. Include an unaltered copy of this License. * I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence. * J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission. * K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein. * L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles. * M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version. * N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section. * O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version. 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements". 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document. 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate. 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title. 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it. 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document. 11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing. ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (C) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

6.3 GNU Lesser General Public License

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below. 0. Additional Definitions.

As used herein, “this License” refers to version 3 of the GNU Lesser General Public License, and the “GNU GPL” refers to version 3 of the GNU General Public License.

“The Library” refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An “Application” is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A “Combined Work” is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the “Linked Version”.

The “Minimal Corresponding Source” for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The “Corresponding Application Code” for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work. 1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL. 2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

* a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or * b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

* a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License. * b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

* a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License. * b) Accompany the Combined Work with a copy of the GNU GPL and this license document. * c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document. * d) Do one of the following: o 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source. o 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version. * e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

* a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License. * b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.