Doyle P. Hunt May 6, 2005

Free Riding in a Peer to Peer Networked Environment

Chapter 1: History of P2P and the Free Rider Problem

The History

Peer to Peer, or P2P, networks, have existed since the mid-1990's, with the Napster phenomenon birth just before Y2K as a marked point in the development of P2P networks. P2P networks are essentially a marketplace for people to share and trade digital files, with little regulation or oversight. As the internet emerged, the ability for any user

and fueled the fire for P2P application development. Napster's success was a product of Network Effects. Networks Effects are a cyclical situation whereby more users provide more content, which in return, attracts more users. This effect quickly spawned a tremendous interest in Napster, which was allowing users to freely share and download music files, which otherwise would have to be paid for or obtained by physically recording a compact disk into a digital recording format. Because of these effects, Napster became by far the cheapest (free), fastest, and most convenient way of building a tremendous music library. This phenomenon has had tremendous ramifications, but also some large benefits, depending on the industry in question. According to the record industry, music sales are down, but at the same time, the large distribution of unprotected copyrighted music led to the huge surge in growth of the MP3 player market. So clearly, P2P networks have had effects that were not foreseen in the early days of these file sharing networks.

As these applications became as commonplace to have on a computer as Microsoft Word, some old-world problems began to hit the P2P application world. 'Network Congestion' and 'Free Riding' arose as two problems that have existed in other social dilemmas that were now harming P2P networks as well. As both of these problems have been realized and studied, solutions have been made to try and forcefully remove the negativities that they may cause, through different application solutions that have tried to e

outlet for outright theft. The current status of the legality of P2P networks is pending in the United States Supreme Court, and a ruling is not expected until June of 2005. Although there are obvious legal issues with infringement upon copyrighted materials, it is very difficult to make a case against the technology behind P2P applications, since the technology itself is not inherently illegal. Very similar legal issues were raised with the Xerox photocopier and Betamax video recorder. Both can be used for illegal means, but the technologies themselves are not altogether illegal, so long as there is evidence that the principal use of the technology does not violate any laws.

The legalities of P2P networks and the decisions that are yet to be made by the Supreme Court and potentially the Legislative Branches of our government, may have profound effects upon the future existence and development of P2P networks. In the current case pending in the Supreme Court, with twenty eight entertainment companies suing Grokster, a P2P application provider, for running a business where its sole profit comes from the exploitation of copyrighted works, although just about everyone agrees that file sharing should be condemned, there is a very broad consensus that the technology itself must not be stifled, otherwise further innovation may be thwarted. Dozens of companies have filed briefs supporting P2P technologies, including Intel, AT&T, MCI, Verizon, and Sun Microsystems along with organizations such as the ACLU, the National Library Associations and the Educational Defense Fund. All of these comp

Copyright Act of 1998 (the DMCA) places the burden on reasonable enforcement upon the owners of the material by stating that,

> "Contracting Parties shall provide adequate legal protection and effective legal remedies against the circumvention of effective technological measures that are used by auth02 T -2 (t) -2 (h) 37.9801 612801]T01 -0.01 -0

quickly picked up (BitTorrent it

Congestion is limited in P2P networks in an ideal situation, since there is no central source for a single file. Ideally, as soon as one user has a file, multiple people can download the file simultaneously, and from there, multiple people can download from each of those peers and so forth. This exponential growth makes reliance on a single peer, such as in a traditional client/server model, much less important, and prevents a single peer or server from having the sole responsibility of providing the requested files. This sharing of responsibility can limit the amount of congestion on a single network or peer. As bandwidth and processing power have increased dramatically as well, connections tend to last shorter amounts of time, and network equipment can handle the requests more quickly as well. One thing that is not greatly changing though is the actual size of files that are being transferred. Music files have typically, and most likely remain, in a compressed form taking approximately $3x10^6$ bytes of disk space. Video however, which has become ever more popular to share after the P2P craze that was started with Napster, takes up considerable more space, but tends to be consumed in much smaller quantities. While some people may consume 2,000 music files on their personal computer, users will typically only have a few movies or television episodes. Nevertheless, the MPAA (Motion Picture Association of America) is taking the lead in prosecuting file swappers, as they feel that the mup

One of the problems the movie industry is facing that is difficult to combat, is the consumer's desire for on-demand availability. Downloading individual episodes of television shows or movies before they are released to dvd or vhs, allows users to have exactly what they want when they want it. Movie studios use windowing as a technique to maximize profits, by staging the release of movies in theatres, home video, and cable television. This demand by consumers and lack of remedy by the Motion Picture Industry has caused file swapping of movies and television shows to be the fastest growing use for P2P networks. BitTorrent, the P2P network that has revolutionized the file sharing methods used for large files has skyrocketed this problem to become a real concern in the industry. One report believes that one third of all internet traffic is related to the BitTorrent network.

Network Congestion

P2P networks go to the extreme by increasing processing power dynamically, by using the resources of a large number of users as a collective whole. As the number of users increases, ideally the amount of processing power and resources increase along with it, however, the *Free Rider* issue can hamper that perfect solution. But even with a small percentage of users with fairly fast connections actually providing their bandwidth, processing power and other resources, P2P networks can provide the same power as very well-built server farms. Problems, such as cracking 40-bit encryption and finding large Mersenne prime numbers (a prim

Free Riding

contributing, but only using the community's resources. This could potentially lead to self-destruction of these file sharing networks that the Movie and Music Industry have so greatly feared.

Users who have been free riding typically do so as a way to prevent viruses, security holes, loss of computing power and criminal prosecution. Some of these problems though are being addressed, and criminal prosecution tends to be the main deterrent from participating as a true peer

centr

uploading or downloading from multiple peers, with a queue of torrent requests, as a way to operate the most efficiently.

According to Bram Cohen,

> "Each peer is responsible for attempting to maximize its own download rate. Peers do this by downloading from whoever they can and deciding which peers to upload to via a variant of tit-for-tat. To cooperate, peers upload, and to not cooperate, they 'choke' peers." [3]

In order to have an efficient algorithm, the resources of connected peers must be reevaluated in a timely fashion. BitTorrent recalculates the current transfer rate every 20 seconds for each connection and reevaluates whether or not it is connected to the best peers every 30 seconds. If the algorithm finds other peers who would create a faster download, it will drop its slowest connection and connect to the new peer in order to facilitate the fastest download. Along with connection speed, the algorithm also uses a 'rarest first' approach, which attempts to download torrents which are least available first, so that its availability will eventually increase.

Lastly, Cohen expresses his interest in distributed collaboration as a way to create a very powerful decentralized service. In order to accomplish this, the system must become pareto efficient. A pareto efficient system is a system in which the system is so optimal, that the only way to further advance any node in the system, could only come at a cost to another node in the same system.

> "In computer science terms, seeking pareto efficiency is a local optimization algorithm in which pairs of counterparties see if they can improve their lot together, and such algorithms tend to lead to global optima." [3]

Clearly, collaboration towards efficiency will ultimately lead to a fast, highly scalable and distributable P2P network.

Gnutella- Initially released in March of 2000 by Nullsoft, a subsidiary of America Online, it was quickly pulled back as it was quickly realized that this application had the same abilities as Napster, which was currently in the process of being shut down by the judicial system. At this time, AOL and Timer Warner were in the process of merging to form AOL Time Warner. As Gnutella had been

Clearly the Gnutella application will work in a small closed environment, but large scale deployment brings about many additional concerns. One of the issues was that peers were trying to contact every peer directly for a request, instead of having request messages forwarded, which could have preser

since the increase in criminal prosecutions of

an individual is acting alone, and when that same individual is part of a group. The most famous story to demonstrate this problem is the *Prisoner's Dilemma*, and this is the story:

"Tanya and CinjCinjrory 1 havrory 1 be 50.4 (s)27 and"Taj 0 Tc ET BT al -0.43yv

really possible to m

infringing upon its copyrights, because the video recorder allowed anyone to time-shift any broadcast and view it at a later time; effectively copying the original broadcast. Sony's defense was that the Betamax video recorder itself was "capab

In this study, a sampling pool of 7,349 peer

provide resources before they can consume, you have a problem when the group is formed, and all you have are new members who cannot consume, because everyone else is also new, and cannot use another's resources until someone uses theirs. If no one is able to consume, than none of the new peers will be able to offer their resources to others. This is a problem with the initialization of Peer Auditing, and one that must be addressed in order for it to function properly. Exceptions must also be made for those who cannot legitimately contribute to the group, and must be given resources first before they can afford to provide resources to other peers.

Chapter 3: A Free-Rider Problem Solution With JXTA

My goal for this thesis

protection. Every group is a sub-network of the *Default Peer Group* network, by which every peer is a member, by default. From this *super group*, all other groups are formed. When a group is formed, a *PeerGroup Advertisement* is created, and propogated through the network. PeerGroups are identified by a unique ID, and can be created by any peer. In order for two peers to join the same group, they must know the unique ID prior to joining the group. Group ID's can be found in one of two ways. First, the peer could have the ID embedded within the application, or it could be obtained from an outside source, such as a website, w

inefficient and un-secure to have to process advertisements belonging to another group, for which a membership has not yet already been created.

Once a group has been formed, and members have been accepted, members will send out information about their means of communication to all of the other members of the group, so that direct connections can be made between any two peers. Since this is a P2P network, all communication occurs directly between the two peers, with no other involvement from

large as 10Mb, and the issues involving larger files stem from the Java Runtime Environment not being able to allocate enough memory to serialize an entire large file at once. When large files (large than the currently available amount of memory) need to be sent across sockets, they need to be serialized into a byte stream, written to disk, and then

When a user joins a group, the user can either provide no, or bogus credentials, and still join a group, if the group does not have any means of authentication. If the group does contain a means for authentication, the user must be first verified before joining the group. The authentication process works as follows.

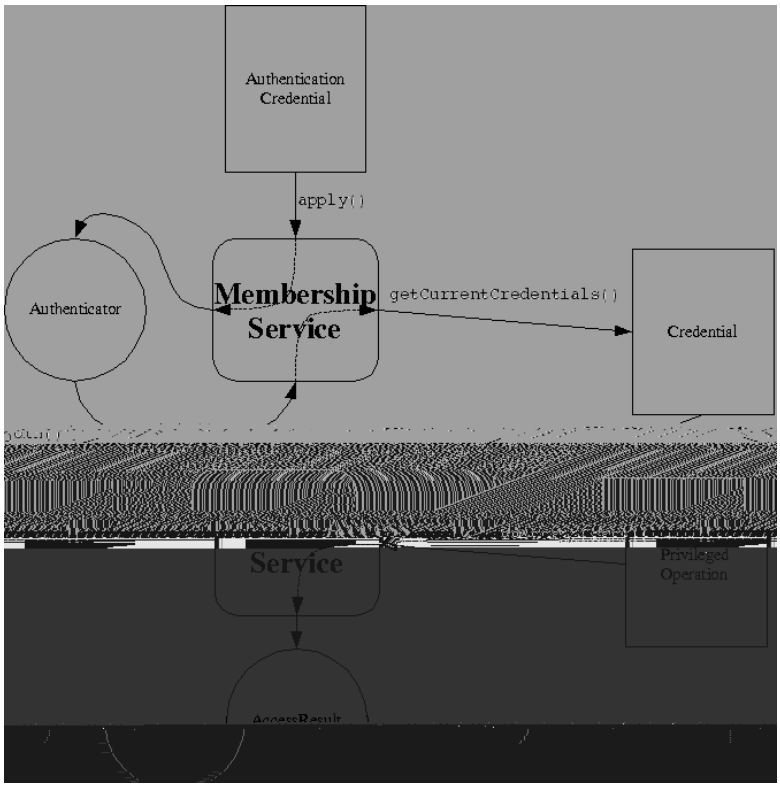

Diagram of an Authentication process in JXTA. (image: API: net.jxta.membership)

focused on large Linux files, and game applications. This later turned out to be a great way to distribute visual media, both copyrighted and non-copyrighted. But if the pool of applications and files were immense, there would be little chance that more than one or two other peers might have what is needed. In this case there must be some other way to negotiate upload and download rates, if it is highly likely that one peer might not have

This PeerGroupID is theoretically unique, and the chances of having another group with the same ID is 1 in $8.7x10^{40}$. So the chances of a collision are pretty minimal. This ID is encoded in the application, and cannot be changed by the user. Only peers who know this ID may join the group. In order to prevent users who do not have the application from joining the group, the GroupAdvertisement is not published, and therefore other peers cannot see the GroupID. If the GroupAdvertisement were published in an XML document, by the DefaultNetPeerGroup's DiscoveryService, any user of any JXTA application would eventually

agreement with the other. They do not leave their cards out for others to take, and verbal communication is essential to making the deal.

Since a user really does not have the ability to free ride, the cost of free riding is immense. The cost would be the inability to use the network or any services it would provide. Since one who free rides would not be able to obtain anything from another peer, unless the other peer made a conscious decision to allow the other peer to free ride on their own file library, it would be a long and arduous process to continually ask others to provide them with something for free. It might work for a short time, but is in no way a sustainable method for using the network. There might be an instance where one peer would not want anything from another peer, but would be willing to share their own library with another. File transfers are made through two separate peers, and broadcasting is not an option, although implementing broadcasting to create group chat or group file sharing would be possible to implement in the application, if such a feature were desired.

This inability to free ride, I believe, would greatly reduce the rate of transfer of copyrighted materials, because of the time and energy required to negotiate a transfer. Only files that were explicitly desired would be sought after, and without a daemon, a full list of available files would not be available for others to browse. One of the problems with other P2P file sharing applications was the ability fo

must come to an agreement with one another, before any file transfer will be made. Instead of the traditional daemon approach, peers *put* files onto another's machine, instead of the requesting peer going and retrieving it. This virtually limits the possibility of any free riding from taking place. Without a central management service, this enforcement must be done from within the application itself. Once the application is launched, it will connect to a predetermined group, and must know of at least one other peer in the group, unless the group is contained within a multicast environment. Once at least one peer, or rendezvous, is located, other peers are subsequently found, and connections between the peers can be made. Once these socket connections are made, any byte stream may be sent from one peer to another, but only after an agreement between the peers has been formed.

Although it is a simple solution in principle, it will help protect the integrity of P2P networks. As other P2P networks grow larger and larger, users feel more obscure, and do not feel as though their lack of participation will go noticed. This is similar to the Voter's paradox, where one vote may not make a difference, but if everyone had the same mentality, there could be no election. Therefore it is imperative that the free riding issue be taken seriously, and directly addressed.

P2P networks have much potential, whether it be for file sharing applications, or distributed applications. The reputation of these networks has been infamous, as the early networks thrived and grew due to the demand for copyrighted me

use. The technology though must be able to develop, as networks of P2P communication would result in efficient network usage, powerful distributed systems, and the ability to not rely on a central server computing model. This technology can be very successful, but only if developers are left unhindered, and problems such as free riding are openly addressed, so as to not cause the self destruction of those systems, and the technology itself.

References

1) Digital Millennium Copyright Act of 1998 – U.S. Copyright Office Summary
http://www.copyright.gov/legislation/dmca.pdf

2) The BitTorrent Effect, *Wired Magazine*
http://wired-vig.wired.com/wired/archive/13.01/bittorrent.html

3) Incentives Build Robustness in BitTorrent, *Bram Cohen,* May 22, 2003

4) http://ntrg.cs.tcd.ie/undergrad/4ba2.02-03/p5.html

5) Free Riding on Gnutella, Eytan Adar and Bernardo A. Huberman
http://firstmonday.org/issues/issue5_10/adar/index.html

6) Kuhn, Steven, "Prisoner's Dilemma", *The Stanford Encyclopedia of Philosophy (Fall 2003 Edition)*, Edward N. Zalta (ed.),
http://plato.stanford.edu/archives/fall2003/entries/prisoner-dilemma

7) Felkins, Leon, "An Introduction to the Theory of Social Dilemmas", Nov 1994.
http://www.spectacle.org/995/sd.html

8) Knowles, Jeffrey, "The Debate over Sony-Betamax and Peer-to-Peer File Sharing", The Computer and Internet Lawyer, Vol. 22, No. 3, pg.1, March 2005

9) Ngan, Wallach and Druschel, "Enforcing Fair Sharing of Peer