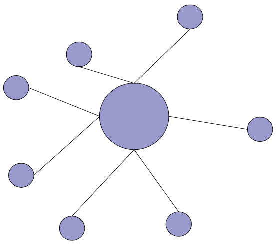
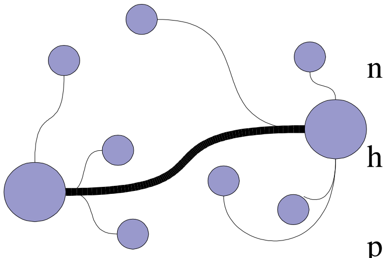
"RIAA sues hundreds more over swapping"[1]. Headlines like this seem to be appearing all the time. The purpose of this paper is to examine the flaws in current Peer-to-Peer networks, and explore some solutions in anonymous file sharing. I will also present a protocol that will provide a reasonably secure way to host files, without revealing your identity. This darknet thus created will be nearly immune from any sort of external control.



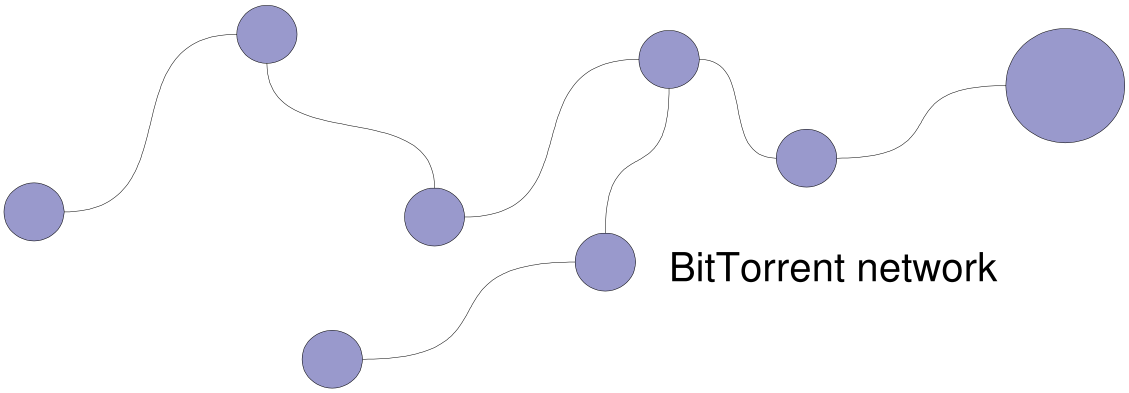
Napster (Centralized)



Kazaa network (Hybrid)

Napster was the first popular file sharing network, but was easily taken down because it relied on central servers[2]. Gnutella solved this problem by creating a completely decentralized network[3], but Kazaa's decentralized network has proven much more popular because it uses a more efficient hybrid model[4]. Still none of these networks offered any sort of anonymity to their users. All that a lawyer needs to do to gather incriminating evidence is to simply search for filenames that his employer does not want distributed, and generating a list of IP addresses hosting those files. In fact they appear to have automated this process, every few months they chose about 500 users, apparently selected a random from theses lists[5]. Though they may not be able to identify a person by their IP address alone, they can subpoena this information from an ISP who will often hand it over.

BitTorent has been the most popular P2P network recently. According to Reuters it accounts for one third of all traffic on the internet[6]. For each file that is hosted on Bittorrent, there must be a tracker to seed that file, and help the nodes that want to add themselves to the chain of downloaders to find where to fit in. Once the tracker goes down, no one can download the file, because all the routing is done by the tracker. BitTorent is an excellent protocol for moving large amounts of data to a lot of people very quickly, but if anonymity is desired, BitTorent is a bad solution, since the IPs of everyone getting the file, and the IP of the tracker are easily obtained[7].



Justin Frankel, the programmer behind Winamp and Gnutella, created WASTE[8], an entirely secure Virtual Private Network that uses public key cryptography. Every user in a WASTE keeps a list of the IP addresses and Public Keys of every user on the WASTE ring he is connected to. All file transfers are encrypted with the public key of the person requesting the file. A WASTE ring may consist of up to 50 people, and currently you cannot connected to more then one WASTE ring at a time.

While WASTE is a fine piece of programming, the main reason why P2P software is so popular is because it gives you the ability to exchange files with people you have never met. An anonymous peer-to-peer network therefore has two competing goal that at first may not seem resolvable. The first goal is the same as any open peer-to-peer network, namely to search as many hosts as possible and exchange files with them in the most efficient way possible. The second goal is not to let anyone know what files you are actually sharing. There are some interesting approaches to attaining these goal, put forward anonymous P2P programs such a MUTE, Freenet and ANts[9]. Both of these use proxies, and try to break files into encrypted segments, so the person who is sending you part of a file is likely not the person who originally hosted the file. In addition not all of the segments are necessary to reconstruct the original file. This is accomplished with error-correction techniques similar to a RAID array of hard drive. The idea is that everyone is hosting some of these encrypted segments, with no knowledge of their contents, so if someone tries to sue you for sending them part of a particular file, you can claim you didn't even know you were hosting that file. They seem to think that this sort of “plausible deniability” will hold up in court, but I disagree.

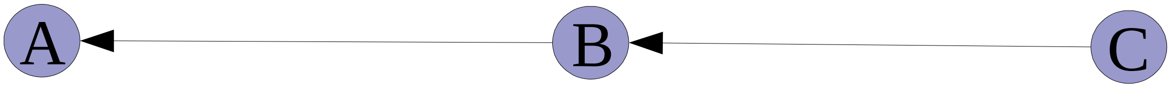
I would propose a network that uses a trade off between these goal to provide better security. Rather than attempting to hide your identity from everyone on the network you will choose a list of trusted friends. These could be people you know in real life, or people you met online. Most users of the network won't really care who is on the list, as long as they are fairly certain the people that they have chosen to trust are unlikely to sue them. As a darknet like this grows more popular, users will have to be more careful who they decide to trust, as corporations and governments will probably use the tactic of winning a users trust to gain access their files and possibly sue them. The benefit of such a system is that each user decides which other users to trust, so it is not necessary to trust unknown hosts.

Using list of trusted friends is similar to a WASTE network, though this darknet will not necessarily have the same “ring” infrastructure because all of the people on your list may not trust each other. Downloads from users that have decided to trust you will be direct, and outside users will not be able to decipher them because they will be encrypted with your public key. The network could be left at that, and the aim of not revealing your identity to anyone you have not specifically trusted would be satisfied, but finding individuals worthy of your trust is a tedious effort once you have gotten past you immediate friends. What you really want is to search the files of millions of users, and it would be impossible manually trust that many without inadvertently selecting a few you really didn't intend to.

The logical to start finding new people to trust would be friends of your friends, and you might actually start doing this to get faster downloads, but it's likely that not all your friends have the same discernment that you do, and may be liable to trusting people that they shouldn't. So in keeping with the concept that each user controls his amount of risk, automatically trusting all your friends friends would be a bad idea. What can be done however, is to use proxies. That is all messages between you and the friend of you friend will pass through your mutual friend.

Let's go for a slightly more concrete example. Suppose that there are three hosts on the network, Alice, Bob, and Carol. Alice and Bob trust each other, and each can access the others files. Bob and Carol trust each other, and both allow the other to freely browse and download their own files.

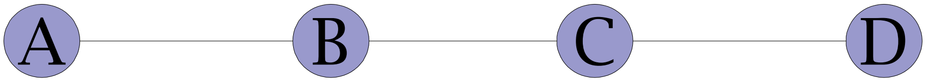
As it stands there to two separate and distinct networks created from these three people. If Alice is looking for a file the Carol has, she will not be able to find it, because in the name of privacy, Bob won't tell her. The solution to this problem is to use Bob as a proxy.



Carol will give Bob a list of her files, and Bob takes this list, removes all references to Carols IP address, and then forwards the list to Alice. Alice can them browse the list of Carols files, which will be identified only by her username, and her virtual network address. Bob's IP address will also be attached to the list so Alice knows who to request the files from. Since Bob trusts Alice, he is not worried about any actions that might be taken against him for the files Carol has decided to host. Searches will work the same way, Alice will request a particular file, and Bob will return results that are hosted on his computer as well as Carol's. If Alice want to download a file that Carol is hosting he will proxy it to her.

It is important that Alice *never* has to communicate directly with Carol, which would end Carol's anonymity. Also all messages (or files) with be encrypted so no outside party can see what is actually happening. Specifically, when Alice requests a file from Bob that is hosted by Carol, this request will be encrypted with Bob's public key. No one but Bob, the holder of the secret key will be able to decipher this request. The request will consist of the filename and the virtual address of where it is hosted. Bob will lookup the virtual address, and not unlike DNS, will match it to an IP address. But instead of giving this IP address to Alice (ruining Carol's anonymity), he will simply encode a request in Carol's public key and send it to her. Upon receiving the request from Bob, Carol will encrypt the file with Bob's public key and send it to him. As Bob is receiving this file he will decrypt chunks of it, encrypt them with Alice's public key and send them on to her. Once the transfer is complete, Alice will be able to decrypt it with her secret key and enjoy the file.

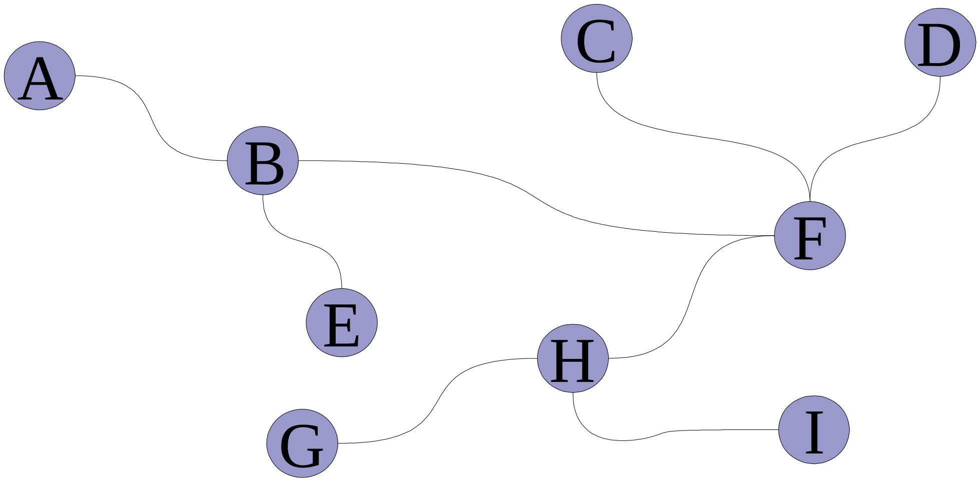
The way to extend this network is to use multiple proxies like Bob to form a chain of trust between users.



Now Alice can get files from Dave by requesting them from Bob, who will in turn request them from Carol, who will in turn request them from Dave. These chains can be as long as necessary, but as they get longer they will becomes more inefficient. One might suppose that in order to access the files of a large enough pool of users the chains would have to be impossibly long.

To answer this question consider a bit of history. In 1967 Stanley Milgrim published the results of his study on social networks as “The Small World Problem” in Psychology Today[10]. He found that there were many relationships between people who had never heard of each other. Though further study he concluded the the average distance between any two people was about six relationships, or six degrees. This principle could be applied to a computer network.

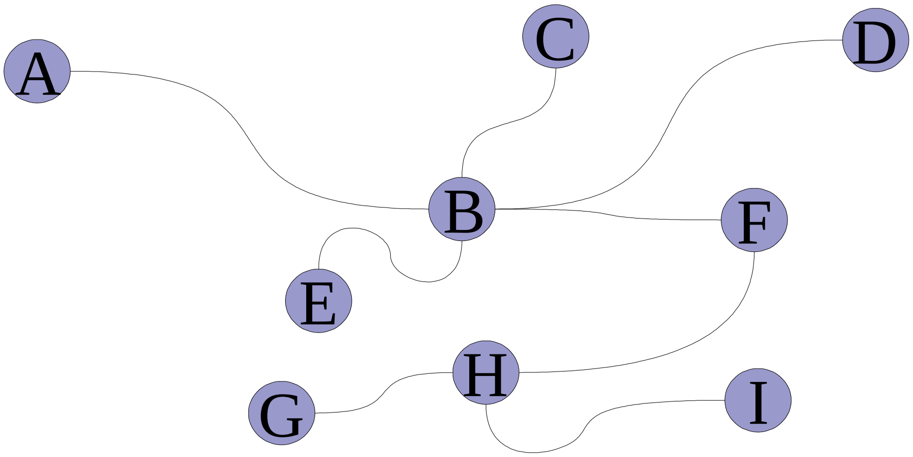
In our darknet, let us assume that one person trusts ten people. Then those ten people each trust ten more unique individuals (that is not each other, not the originating node, and not the same ones as each other). If this process is repeated six times, there will be a million nodes, each one no more that more than six relationships from each other. Even after averaging the rest of the network, according to Milgrim's studies, any user should be an average of six relationships from any other user.



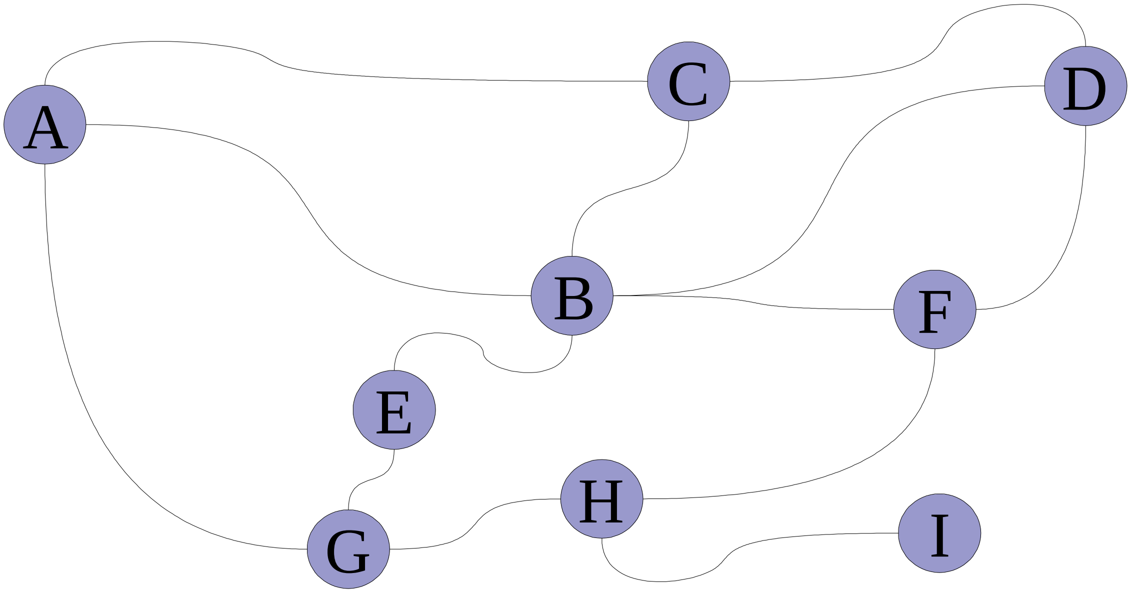
If you do have to proxy a file though four extra people (besides you and the person you are downloading from) transfers are likely to be slow and unreliable. In fact one person may go offline and break the chain, but since you have the virtual address of the file you can continue to search for a new chain that connects you to that person. But in practice chains will probably be short. Because you are trusting all your friends directly there are no proxies used when you download a file from them. And because they are your friends they probably have similar interests as you, so they will have files you want. Also popular files will be hosted by many people, and one of those is likely to be one or two degrees from you. So obscure files will be nearby because people who have similar tastes will trust each other, and popular files will be nearby simply because they are so widely distributed. The only files that will require long chains are obscure files that your friends do not want, or that are new to the network and your friends do not yet have.

Creating these chain will involve routing algorithms comparable to those currently used to find the least-cost path between two hosts on the internet. There will likely be multiple paths between two hosts and the on with the fastest transfer will be chosen. Also there are likely to be multiple hosts with the same file. In this case part of the file may be transferred from first host, while another part is gotten from the second.

Some requirement for this sort network to work well are always on broadband connections, and large amounts of hard drive space to cache all the file being proxies through your node. I don't believe these present major problems, as current P2P networks are already populated with broadband users with large hard drives. One problem that may occur is something Milgrim described as funneling. In his studies of social networks he found that one extremely well connected person might be the only connection between many otherwise disconnected people.



In this figure almost all file transfers are routed through Bob. Bob being a very sociable person has decided to trust many of his friends. The problem is that many of Bob's friends are are paranoid hermits who have only begrudgingly trusted Bob in order to gain access to the darknet. Bob does not enough bandwidth to efficiently proxy all the downloads that people are trying to route though him. Also if Bob goes off line, his anti-social friends will have no way to access the rest of network. The solution for this problem is for Bob to convince more of his friends to trust each other.



This will give the network redundancy it needs, giving his friends faster downloads, all while taking a load off off Bob. Being a social network, you will have to convince other real people to trust you. A consequence of this structure is that social engineering will be the best attack on the network. An easy way to do this would be to hang out on forums where the darknet is discussed, pose as newbies and beg for someone to trust you so that you can get on the network. Once someone does trust you (exchanging public keys), you can then see all the files that that person is sharing and sue them for everything they have. However once you have done that you will have to snag another person willing to trust you. You trustworthiness will be seen as zero, since the network is anonymous, no one will be able to tell what you did last time, but if instead of immediately suing the first person to trust you, you decide to infiltrate the network, trusting many people, even become a funnel transferring files between subnetworks, you could come up with quite a list of people to sue.

This proxying process used by this network will by definition inefficient, in a worst case scenario, you will have to download 600 megabyte, and upload 600 megabytes before you can get 100 megabytes for yourself. But each user will have to decide how many people can trust, because assuming all trust are reciprocal, each person he trust will bring him on degree closer to many people.

As P2P networks become more sophisticated attack on them will have to become more sophisticated and intrusive. One forum member pointed out that though outside parties might not be able to tell what sort of data you are passing around they will be able to tell that you are passing around a lot of encrypted data, and may even be able to tell what program you are using to do it with. (This packet-shaping has clearly showed us). If such networks were made illegal, government agents could easily find the home of people running the program and confiscate their computer, possibly using information extracted from it to implicate their friends. Such Orwellian tactics are not likely to bear well with the general population however, and though they have been ways suggested to to prevent people you are are even running the program, such a extensive use of stenography, I don't think they will ever very important to the network.

I have no illusions that authorities will find ways to monitor the content of darknets, but they will find it difficult if not impossible to control that content. Indeed that is how the protocol is designed, it is easy to gain access to a lot of information, but difficult if not impossible to determine who is hosting it.

If lawsuits continue, and are successful in decreasing the use of current insecure P2P networks, I believe anonymous networks, or darknets will rise in popularity, and I believe a protocol like this one will be used for one of them.

References

1. MSNBC “RIAA sues hundreds more over swapping” 19 Nov. 2004. 5 Dec. 2004.
<http://msnbc.msn.com/id/6531127/>

2.“Napster” *Wikipedia - the Free Encyclopedia.*
16 Nov. 2004. 30 Nov. 2004.
<http://en.wikipedia.org/wiki/Napster>

3.“Gnutella” *Wikipedia - the Free Encyclopedia.*
25 Nov. 2004. 30 Nov. 2004
<http://en.wikipedia.org/wiki/Gnutella>

4.“KaZaA” *Wikipedia - the Free Encyclopedia.*
30 Nov. 2004. 30 Nov. 2004
<http://en.wikipedia.org/wiki/KaZaA>

5. Rohrer, Jason“How File Sharing Reveals Your Identity” 5 Dec. 2004
<http://mute-net.sourceforge.net/howPrivacy.shtml>

6. Pasick, Adam. “File-Sharing Network Thrives Beneath the Radar” *Reuters*
6 Nov. 2004 2 Dec. 2004
<http://www.reuters.co.uk/newsArticle.jhtml?type=technologyNews&storyID=6734497&section=news>

7. “BitTorrent” *Wikipedia - the Free Encyclopedia.*
30 Nov. 2004. 30 Nov. 2004
<http://en.wikipedia.org/wiki/BitTorrent>

8. “WASTE” *Wikipedia - the Free Encyclopedia.*
19 Nov. 2004. 30 Nov. 2004.
<http://en.wikipedia.org/wiki/WASTE>

9. Ingram, Michael. “ANts P2P2P: A New Approach to File-Sharing” *Slyck News*
13 Sep. 2004. 30 Nov. 2004.
<http://www.slyck.com/news.php?story=567>

10. “Small world phenomenon” *Wikipedia - the Free Encyclopedia.* 30 Nov. 2004. 30 Nov. 2004. <http://en.wikipedia.org/wiki/Small_world_phenomenon>

Endnotes

Another application for this sort of web of trust would be an email system resistant to Spam. The Spam problem could easily be fixed by using white lists, and only accepting e-mails from friends that you trust, much like WASTE only allows trusted friends to connect, but sometimes it is desirable to receive e-mail from a person that you don't know. Heres how it would work: first you set up a whitelist of you friend form whom you will always accept e-mails from. Then if a e-mail comes from a friend of one your your friends you will accept it. In fact you will accept any message that comes through a chain of trusted connections leading to you. Even though you may not know many people in the chain, it will be enough that they all trust each other not to be spammers. All messages must be digitally signed by someone in order to be delivered. If someone begins spamming other people his friends will revoke their trust in that person, and he will not be able to send messages to anybody. In e-mail end-to-end encryption will be much more important than anonymity, so all your “friends” won't be able to read your messages.