Improving Gnutella Protocol:

Protocol Analysis And Research Proposals

Igor Ivkovic
Software Architecture Group (SWAG)
Department of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1 Canada
+1 (519) 888-4567 x3088
iivkovic@plg.uwaterloo.ca

ABSTRACT
This paper presents an analysis of the Gnutella protocol, a type of the peer-to-peer networking model, that currently provides decentralized file-sharing capabilities to its users. The paper identifies the open problems that are related to the protocol and proposes strategies that can be used to resolve them.

Initially, the paper explains the basics of the peer-to-peer networking, and then compares the two types of this networking standard: centralized and decentralized. The Gnutella protocol is classified as a decentralized model, and its characteristics and specifications are described accordingly.

The issues that are outstanding for the protocol are listed and clarified through observation. Based on the observed irregularities in the Gnutella's behavior, two different strategies for the problem resolution are proposed and the schemes for the user registration, user rating and establishment of the Gnutella sub-networks are proposed. For each of the strategies, the future steps in realization are also specified.

Keywords
Gnutella protocol, peer-to-peer networking, P2P, levels of anonymity, user registration, user rating, distributed networks, improving download success, improving the reach of queries

1 INTRODUCTION

In early 1999, Shawn Fanning, an undergraduate student at the Northeastern University in the United States, started a phenomenon called Napster [1]. Fanning envisioned Napster as a service that allows users of his system to list the MP3-encoded music files that they are willing to share and let other users download them through the Napster network. The central computer would then at all times have an up-to-date master list of files that people are willing to share, and the list would be updated by the users' software as they log on and off the system.

Fanning's idea required an network infrastructure of servers for the centralized peer-to-peer data access and data storage, and a corresponding bandwidth for allowing a large number of user connections. After his idea is implemented, Napster at one point attracted over 30 million users with over 800 thousand of them accessing the network simultaneously [1]. The upper limit on the number of users of Napster is only imposed by the bandwidth, as the service itself never lacks popularity.

Unfortunately, due to violations of the copyright laws and by the order of the Supreme Court of the United States, Napster is forced to impose complex limitations on the shared files thereby downgrading the quality of their service. Finally, in July of 2001, Napster is forced to shutdown its servers due to software-related problems that occurred as a result of Napster's developers trying to ensure that these limitations are imposed throughout their network.

Beside the copyright infringements, the reason why the Napster network fails to provide quality-of-service (QoS) to its users is due to its centralized peer-to-peer character. Since Napster has a single point of entry, the network can completely collapse if its central point becomes incapacitated. In addition, this central point has a complete authority over the data distributed through the network and is solely responsible for its contents (Napster's legal problems stem from this fact). A solution to providing QoS in the peer-to-peer environment is by using a decentralized model instead, and having multiple access points, some of which if incapacitated would not incapacitate the entire network. One such decentralized peer-to-peer model is the Gnutella network based on the Gnutella protocol.

2 WHAT IS PEER-TO-PEER NETWORKING?

Before analyzing the details of the Gnutella protocol, the reasons why peer-to-peer networking is important and why people are considering peer-to-peer file sharing for anything else but sharing of copyright-infringing music and video files must be analyzed.

As mentioned before, Napster is a type of the peer-to-peer networking model in which each party has the same communicational capabilities and either party can initiate a communication session. On the Internet, peer-to-peer (P2P) is a type of networking that allows users with the same networking program to connect with each other and directly access each other's files.

Business advantages of using peer-to-peer networking are still being discovered [2], and the key returns are harnessed through

- Distributed processing, that is allowing users of the network to schedule batch-jobs that are processed by the computers on the network during their idle time thereby decreasing the need for new computing resources; and
- File sharing, that is allowing users to exchange data directly without storing files on a centralized server thereby avoiding the need to establish a centralized server and allowing two businesses to communicate with each other directly.

The future potential for P2P lies in its ability to change the structure of the Internet [1]. By allowing users that are accessing data from a particular web site to retrieve the cached copy of the data from a geographically closest point to their location would mean a change from the web-site-centric to the purely distributed data distribution model for the Internet.

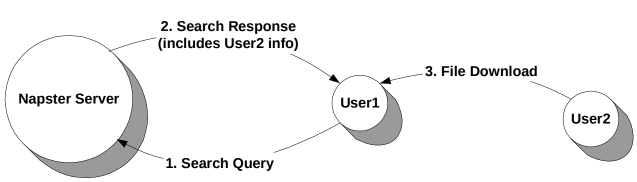

Figure 1: Centralized P2P Model

When it comes to P2P file sharing specifically, there are two models: centralized and decentralized. The centralized model, used by Napster, uses a central system that directs traffic between individual users. The central servers maintain directories of the files shared by the users of the system. The servers allow query of their database and provide results that allow a user doing a query to establish a direct connection (typically using HTTP) with a user who is sharing a desired file. Figure 1 further clarifies this model.

The key advantages of the centralized model are as follows [3].

- The existence of a central index that allows users to search and find the desired files efficiently.
- The mandatory registration of all users of the system thereby ensuring that each file query reaches all users currently connected to the network.

The key disadvantages of the centralized model are as follows [3].

- The single entry point that creates a possibility of the entire network becoming incapacitated if this main entry point is disabled (e.g., Napster's failure under legal pressures).
- The invalid data in the database resulting from the periodical, not real time, refreshes of the data in the database.

The decentralized model respectively holds advantages and disadvantages that are contrary to the disadvantages and advantages of the centralized model. For example, having multiple points of entry in the distributed model prevents the scenario in which the entire network is disabled due to one or more of its entry points becoming incapacitated. Another example is the lack of registration in the decentralized network thereby decreasing the breadth of data being queried and not providing the users with the same QoS as in the centralized model.

The decentralized P2P file-sharing model is employed by the Gnutella protocol, so more details on this model are presented as part of the Gnutella protocol analysis.

3 WHAT IS GNUTELLA?

Gnutella is a decentralized P2P file-sharing model developed in the early 2000 by Justin Frankel's Nullsoft (AOL subsidiary and the company that created the WinAMP MP3 player) [4]. Gnutella's development was halted shortly after its results were made public, and the actual protocol was reverse engineered using the code that was downloaded from the Nullsoft's web site just before its closure. Today there are numerous applications (referred to as Gnutella clients) that employ the Gnutella protocol in their own individual way and that allow their users to access the Gnutella network.

To share files on the Gnutella network, a user (node A for example) starts with a networked computer that runs one of the Gnutella clients. Since this node will work both as a server and a client, it is generally referred to as a (Gnutella) "servent" (both a SERVer and a cliENT). Node A will then connect to another Gnutella-enabled networked computer (node B for example) and then A will announce existence to B. Node B will in turn announce to all its neighboring nodes (nodes C, D, E, and F for example) that A is alive. This pattern will continue recursively with each new level of nodes announcing to its neighbors that node A is alive [4]. Once the node A has announced its existence to the rest of the network, the user at this node can now query the contents of the data shared across the network. Figure 2 further clarifies this model.

This announcement broadcasting will end when the Time-To-Live (TTL) packet information expires; that is, at each level the TTL counter will be decreased by one from some initial value until it reaches zero at which point its broadcasting will stop. To prevent users from setting this initial TTL value too high, the majority of the Gnutella servents will refuse packets with excessively high TTL value. However, from the users perspective, maximizing the chances of finding the required file means using as high as possible TTL value therefore creating a trade-off point for this network. Where low TTL means minimizing the usage of the network resources, high TTL value means maximizing the QoS provided to the users of the network. The optimal TTL value would then (among others) depend on the network topology and traffic characteristics for a particular location and a particular time of the day, respectively, when the query is done.

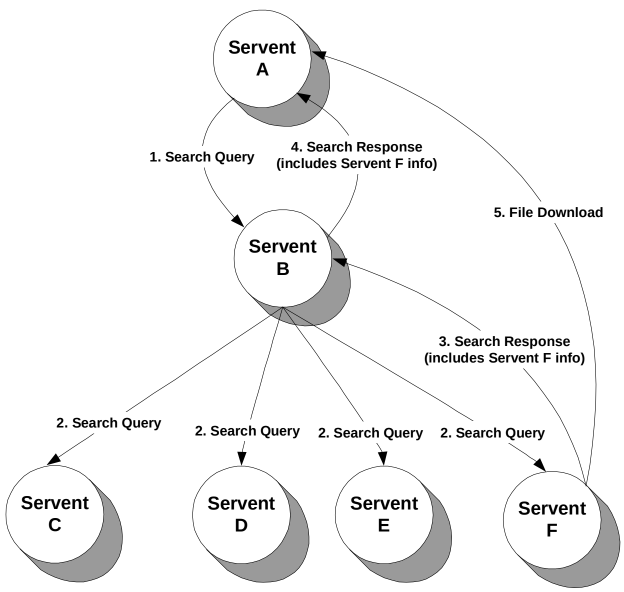

Figure 2: Gnutella Decentralized P2P Model

Gnutella Protocol Specifications [5]

All Gnutella communication happens on top of the TCP/IP protocol. Once a TCP/IP connection is established between two servents, the Gnutella connection string "GNUTELLA CONNECT/<protocol version string>\n\n" may be sent by one of the clients (the current protocol version string is "0.4"). The servent responding to this connection may respond with a "GNUTELLA OK\n\n" message thereby establishing a valid Gnutella connection between these two servents. Any other response to the original connection string will be taken as a communication-rejection by the initiator servent.

After a connection is established, two servents communicate with each other by exchanging Gnutella protocol descriptors. Gnutella protocol also defines the rules for how these descriptors are exchanged between nodes. Figure 3 gives an overview of Gnutella protocol descriptors.

Descriptor	***Description***
Ping	*Used to actively discover hosts on the network. A servent receiving a Ping descriptor is expected to respond with one or more Pong descriptors.*
Pong	*The response to a Ping. Includes the address of a connected Gnutella servent and information regarding the among of data it is making available to the network.*
Query	*The primary mechanism for searching the distributed network. A servent receiving a Query descriptor will respond with a QueryHit if a match is found against its local data set.*
QueryHit	*The response to a Query. This descriptor provides the recipient with enough information to acquire the data matching the corresponding Query.*
Push	*A mechanism that allows a firewalled servent to contribute file-based data to the network.*

Figure 3: Gnutella Descriptors Overview

Given that Gnutella protocol is positioned on top of the TCP/IP stack, the IP addresses in the IPv4 format are used as part of the Gnutella descriptors for the node identification (e.g., when establishing downloads between two servents). In the IPv4 format, the address 208.17.50.4 would be represented as follows.

0xD0	*0x11*	*0x32*	*0x04*
Byte 0	*Byte 1*	*Byte 2*	*Byte 3*

It also must be noted that the header in the format as shown in Figure 4 precedes each Gnutella descriptor. In this header, "Descriptor ID" stands for a 16-byte string that uniquely identifies the descriptor on the network while "Payload Descriptor" field identifies the type of the descriptor (i.e., 0x00 = Ping, 0x01 = Pong, 0x40 = Push, 0x80 = Query, 0x81 = QueryHit). The TTL counter, as described before, and the Hops counter, that stands for the current number of network hops that this descriptor has done, must conform with the equation TTL(0) = TTL(I) + Hops(I). That is, at each network hop, the TTL counter must be decreased by one while the Hop counter must be increased by one. As stated before, the transfer of this descriptor stops when the TTL counter reaches zero.

	Descriptor ID	Payload Descriptor	TTL	Hops	Payload Length
Offset	*0 15*	*16*	*17*	*18*	*19 22*

Figure 4: Gnutella Descriptor Header

As for the Gnutella descriptors, the following information is relevant. For more detail on the descriptors one can consult "The Gnutella Protocol Specifications" [6].

- Ping descriptors
  - Hold no associated payload and are of zero length
  - Used for probing the network and when responded to with a Pong descriptor a connection between the Ping and Pong senders is established
- Pong descriptors
  - Used as replies to Ping descriptors
  - Carry the following network load

	Port	IP Address	# Of Files To Share	# Of KB Shared
Offset	*0 1*	*2 5*	*6 9*	*10 13*

- Query descriptors
  - Used for querying the network for a particular file or files
  - Carry the following network load

	Minimum Speed	Search Criteria
Offset	*0 1*	*2 …*

- QueryHit descriptors
  - Used for positive "file found" replies to a query
  - Carry the following network load

	# Of Hits	Port	IP Address	Speed	Result Set	Servent ID
Offset	*0*	*1 2*	*3 6*	*7 10*	*11 n*	*n n+16*

  - "Result Set" field is of the following form

	File Index	File Size	File Name
Offset	*0 3*	*4 7*	*8 …*

- Push descriptors
  - Used for getting files from Firewall protected servents
  - Carry the following network load

	Servent ID	File Index	IP Address	Port
Offset	*0 15*	*16 19*	*20 23*	*24 25*

The Gnutella protocol rules also indicate for all descriptors that a particular response message can only be sent along the same path that carried the matching request-for-response descriptor. For example, Pong descriptor can only be sent along the same path that carried the incoming Ping descriptor, and the analog holds for QueryHit and Query, and Push and Query descriptors respectively. Ping and Query descriptors are broadcasted to all neighbors, and the servent that is recognized as the target of a particular descriptor will not forward that descriptor further. The downloading process is done using the HTTP protocol scheme based on the information extracted from the "Result Set" field from the QueryHit descriptor.

Gnutella Clients Overview

To access the Gnutella network, a user must have a Gnutella client. Given the open-source nature of this protocol and its increasing popularity, there are many different Gnutella clients available today. Among the breadth of these clients, the more popular and the more established are BearShare, available only for Windows; Furi / Phex, available for Windows, UNIX / Linux, and Macintosh; LimeWire, available for Windows, UNIX / Linux, and Macintosh; and ToadNode, available only for Windows.

Since the Gnutella protocol is reverse engineered and since its source code is publicly available in the form similar to open-source, most of these clients have employed and extended the protocol to suit their needs. Some of these modifications serve various business models that are behind the software, but most of these extensions are meant to deal with the outstanding problems in the Gnutella network.

Gnutella Research Directions

Presently, the work done on improving Gnutella is focused on the following areas of problems related to the Gnutella client [6]. These problems mostly stem from the fact that Gnutella was released to the public without being properly reviewed and tested with many significant issues being integral to the protocol design itself.

- Download failures – This problem is the biggest reason why many users have given up on Gnutella. It was addressed soon after the initial release of Gnutella, but still remains unresolved.
- Scalability – Many people in the scientific community who have a brief knowledge of Gnutella say "the protocol does not scale". The problem refers to the largest responsive section of the public Gnutella network and its size limitation.
- Fragmented development – Due to the lack of properly organized and structured Gnutella development (e.g., many Gnutella clients implement different non-standardized changes to Gnutella to solve the same problem), many efforts on how to improve this

protocol never got past the proposal stage.

- Encouragement of content sharing – Promoting and ensuring content sharing in general and content sharing of "complete" files. This problem is closely related to the downloading failures problem where people list files as available for download, but do not allow other users to get them from their computers by limiting the upload speed.
- Reducing browsing downtime – Blocking or partially blocking people who are just browsing the network without sharing any files on the account that such people unjustifiably use most of the upload slots on the Gnutella clients.
- Reducing unnecessary network traffic – Among others, deciding which parts of the protocol should be broadcasted through the network. As stated previously, all Ping and Query descriptors are currently broadcasted through the network.
- Creating and maintaining a healthy network structure – Among others, using rebalancing, different TTL strategies, and prioritization of requests.
- Addressing security concerns – Higher bandwidth users with static IP addresses who employ Gnutella clients are exposed to various security threats that are currently almost not addressed at all.

4 RESEARCH PROPOSALS

After using and analyzing the LimeWire Gnutella client on the Windows platform and comparing this client to other P2P file-sharing clients based on similar standards (e.g., BearShare, Morpheus, Freenet), the following observations are made.

Downloading failures that are frequent with Gnutella are significantly degrading the quality of service provided by this network. In 100 random searches with TTL value of 10 based on some of the most frequently used keywords of non-adult content [7] and personal interests, the following holds.

- Over 95% of all searches failed to provide results that exhibited
    - Reliability (i.e., would not become inaccessible during the download),
    - Appropriate downloading speed (i.e., in this case, this speed is assumed to be over 30 KB/sec for the Cable/DSL users that can utilize 100 KB/sec and higher downloading speed),
    - Complete data (e.g., for an audio file, a complete song as published by the author of the song), and
    - Easily comprehensible results (i.e., results returned are generally hard to understand and all use very inconsistent naming strategies).
- The LimeWire does provide "Quality" rating for each of the results, but this rating is not relevant in many cases as the "higher quality" sites still exhibit the similar problems as the "lower quality" ones.
- The Morpheus service provides higher downloading speed in most cases thanks to its "file segmentation" feature that splits the file among the identified hosts that offer the file and assigns corresponding percentages of the file to these hosts based on their access speeds (i.e., higher access speed hosts are assigned a larger data segment).
- The amount of accessible data in the Gnutella network peaked at 18 TB, according to the LimeWire status bar, which exceeds the amount of data available in other services such as Morpheus therefore possibly indicating the higher popularity of the Gnutella network.
- The ability to control uploading and downloading speed for the client is extremely valuable for the users on higher-bandwidth nodes and is provided by the LimeWire Gnutella client.
- The user registration service that is provided as part of Morpheus is not available in Gnutella. The Morpheus' registration contributes to the higher breadth in data searches, analog to user registration advantages in centralized P2P networks as mentioned above.
- Anonymity of the network users that is provided through the Freenet service is not available in the Gnutella network where each user's IP address is broadcasted publicly.

These observations indicate the acuteness of the problems still unresolved in the Gnutella network. Two problems that seem to degrade QoS the most are the "Download failures" and the "Encouragement of content sharing" while the other problems are also directly or indirectly linked to these observations.

In order to resolve these problems the following projects are proposed.

Establishing User Registration And User Rating System

To improve the reach of the user queries and the download success probability and to create a possibility for indexing the network, user registration and user rating system must be established. This system should be based on Flinn and Maurer's paper from 1995 on "Levels of Anonymity" [8] where the user anonymity is defined through 6 levels of identification ("level 0" being the complete anonymity and "level 5" providing non-repudiation in identification). Currently, all the users of the Gnutella network are at the "level 1" of this system (i.e., are identified by their systems based on their IP addresses).

User Registration

To establish an identity for the user of the system, "level 2" identification or higher is required (i.e., per-name identification). For example, based on the Napster service, each user is assigned a username and a password, and required to provide a valid email address and their geographical information. Geographical information can be used to improve the topology of the network (i.e., classify the results based on the geographical closeness of two servents).

For Gnutella, given the distributed nature of the system, one cannot centralize this identification. Instead this scheme should be added on the protocol level as follows (see Figure 5 for clarity).

- When registering for the network, a user is required to register and to provide the registration information that includes
    - Pseudonym
    - Password
    - Geographical location
    - Email or real-time messenger ID

      (e.g., ICQ number)
- The registration process is handled by one of the registration servers that hold the distributed database that is synched over night. If one of these servers is incapacitated, the others could take its place and the status of all servers is always available. By registering, a user's pseudonym is assigned to their IP addresses and this relation can only be updated by a user who holds the corresponding username / password combination. However, this information updating will be available as an automatic option in the Gnutella client thereby providing convenience for the users with dynamically changing IPs.
- After registration, a user is assigned a unique ID that becomes a part of the node information returned about the user's servent. Each search request from the user and each search that returns a "file found" response from the user's servent would return this unique ID as a part of the Query or QueryHit descriptor, respectively. This ID would then provide more information about this user such as their rating and their contact information.
- As in the Napster network, the rating and the contact information could be provided by the registration servers while returning the search results (slows down the search) or when trying to download a file from this user (slows down the download), and this choice is available to users from their Gnutella clients.
- Having the user's rating and contact information allows others to contact this user about the file transfer questions and requests, or to confirm the rating of this user and make their file transfer decision based on this rating.
- Load balancing could pose a significant problem when it comes to handling user registration through registration servers. Some of the ways of dealing with this issue could be through balancing loads depending on the (1) geographical location (e.g., servers closest to the registrant could provide fastest access) and (2) time of the day (e.g., servers in Europe could be less busy during the European night time), or through (3) a customized load balancing function that will take into account all relevant variables for the corresponding servents.

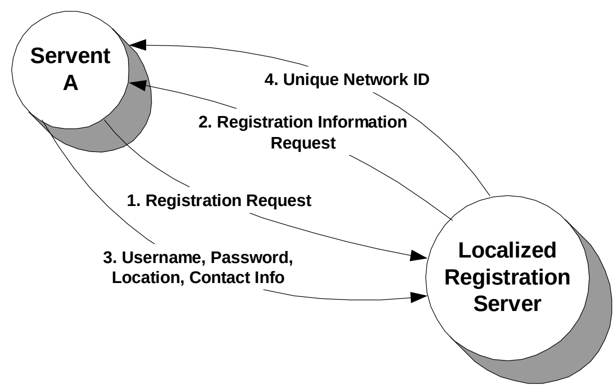

Figure 5: User Registration Scheme Overview

The research problems with the identification scheme are

- Security – Developing further the security scheme proposed here so that malicious clients cannot compromise identity of the Gnutella users.
- Registration server load – Developing further the load-balancing scheme proposed here so that the registration servers are not a bottleneck for the Gnutella users.
- Full utilization of the registration information – Developing further the ideas given here on how to utilize the registration information to the full extent. The possible directions are as follows.
    - Indexing files for each user that registers on the network and having these file lists available on the registration. Or, developing localized searching servers where the same load-balancing problems would apply.
    - Caching queries for the users in the same

"neighborhood" based on the users geographical location.

- File segmentation analog to the file segmentation for the Morpheus service where the file is segmented among the hosts in the local or semi-local area and then concurrently downloaded from all locations at once thereby improving the download speed.
- Prioritizing the query results based on the load balancing requirements.

The implementation details include extensions to

- The protocol that would support registration, and
- The Gnutella clients that would support registration and allow users to harness QoS benefits provided by these extensions.

User Rating
Concurrently with the user registration, the same infrastructure would be used to establish and maintain users' ratings. During registration, user's initial rating (e.g., 0 indicating neutrality) would be established. Through time, after each successful download from the user, the user's rating would increase based on the size of the file transferred and based on the feedback rating provided by a user who downloaded a file in question. After each failed file transfer, based on reasons behind it and feedback left by a user who was attempting the transfer, the file-providing user's rating would be penalized accordingly.

Moreover, there could be two rating schemes where one rating schemes indicates a feedback rating provided by the users after each successful or unsuccessful file download, and another one indicating the amount of data successfully transferred from the user. Two ratings could then indicate how willing and how capable, respectively, is the user in question to share files.

To encourage users to improve their rating, the Gnutella clients could establish a scheme where the downloading would not be allowed for the users with the negative rating and the bandwidth priority is given to those users with the higher rating. For example, for three users with respective ratings of –5, 4 and 6, the bandwidth of 100 KB/sec would be assigned respectively among these users as 0, 40 and 60 KB/sec.

The rating system is employed by various other services such as online bidding (e.g., eBay.com), and online file exchange (e.g., bulleting board systems), and has created a type of an honor system where users provide good service and ask for good feedback in return thereby increasing their chances to sell an item (on online bidding systems) or to find a file of their choice (online file exchanges).

The research problems with the rating system would include the following.

- Rating scheme – Further developing the rating scheme given here and ensuring the meaningfulness of this scheme.
- Full utilization of the rating information – Further developing the ideas presented here on how to utilize the rating system to resolve various other problems in the system. For example, some of the possibilities are.
  - Resolving incomplete data problem by incorporating additional rating points for all users who share complete files. The users who downloaded the files in question would judge their completeness.
  - Dealing with "free loaders" (i.e., users who share no data or who share meaningless data) by either penalizing them for having no data in their shared file list or for having a neutral (no downloads) rating for a prolonged period of time. Also incorporating all malicious schemes that could be established to negate any solutions that might be established to deal with this problem.

The implementation tasks would be analog to the ones listed for the user registration (i.e., extensions to the protocol and to the clients to fully support and utilize this scheme).

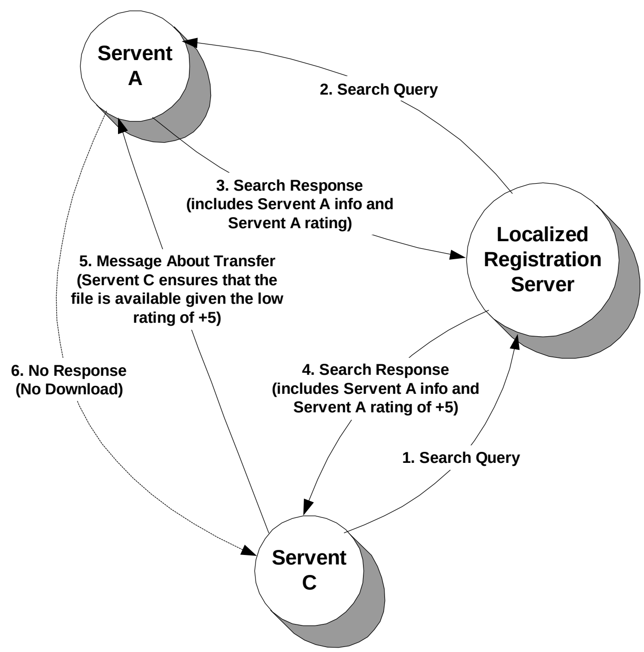

Figure 6: User Registration And User Rating Schemes

Establishing Gnutella Private Networks
After the implementation of the user registration system, the possibility of providing private networks as parts of the Gnutella network is created. By providing higher-level identification (e.g., "level 3" and higher), one can establish

the Gnutella private networks with the higher degree of security.

By establishing specialized registration servers that would be in charge of these sub-networks, various companies and organizations could utilize the Gnutella protocol to establish their individual file sharing networks. For example, high security Gnutella network could be used for redundant file backups across all machines without the need for en expensive backup server. Moreover, if the scheme for the distributed processing is established, these organizations could also utilize the Gnutella protocol for their distributed processing needs.

The proposed scheme for this extra service is a slightly modified version of the general registration scheme where a specialized registration server would handle all registration for the users of the sub-network. Higher-level encryption would be used for communication between the registration server and the servents, and the servents themselves would communicate on a port specifically dedicated to their network. All the ratings for the nodes could be ignored (set to some default value) due to their irrelevance of the user registration for these sub-networks.

Research problems for this idea include choosing a security scheme for communication between the servents and the registration server (e.g., firewalls, VPN, or IPSec), and a scheme for localized file indexing (e.g., by sharing a distributed and on-demand updated file index among the members of the network). The implementation involves extending the registration and the rating scheme implementations to suit this new Gnutella network feature.

5 CONCLUSIONS

The Gnutella protocol is a P2P networking model that thanks to its decentralized nature holds many advantages over competing centralized P2P models such as Napster. For instance, question of reliability cannot be answered in centralized P2P networks since these networks are dependent on their central access point, which if disabled incapacitates the entire network. In contrast, distributed models such as Gnutella have many access points and are more difficult to incapacitate if one or more of its access nodes are disabled.

In general, P2P networking has two main applications: distributed processing and decentralized file sharing. Many companies such as Intel and Microsoft are starting to realize the benefits of the peer-to-peer networking, so it is said that P2P file sharing holds great potential for the future. In addition, P2P networking could change the structure of the Internet from the web-centric to the purely distributed data model by allowing users that are accessing data from a particular web site to retrieve the cached copy of the data from a geographically closest point to their location.

Given its many problems, the Gnutella protocol does not provide the required quality of service for rapid popularity growth, which was a characteristic of the Napster network. Mainly due to the downloading failures and unwillingness of the Gnutella users to share their files through the network, many new users of Gnutella are turned back and are forced to find other P2P file sharing options. Through observation and analysis, it was also realized that a typical search on the Gnutella network fails to provide results that are reliable, that give appropriate downloading speed, that provide complete data, and that are in easily understandable form.

In order for the protocol to establish itself among the individuals and the organizations as a file sharing method of choice, it will have to go through many changes to resolve its internal problems. One possible solution to some of these problems is the establishment of the user registration and user rating system that should provide – among others – improvements in the file downloading, the encouragement of content sharing, the sharing of complete content and the establishment of new features and services.

REFERENCES

1. “Testimony of Shawn Fanning“. Online. Napster Corporation. http://www.napster.com/pressroom/001013.html. October 2000
2. Gomes. "Is P2P plunging off the deep end?". Online. WSJ Interactive Edition. http://www.zdnet.com/zdnn/stories/news/0,4586,2704598,00.html. April 2001
3. Cho. “Understanding Peer-to-Peer Networking and File-Sharing”. Online. LimeWire Corp. http://www.limewire.com/index.jsp/net_improvement. 2001
4. Gomes. "Gnutella keeps growing and growing". Online. WSJ Interactive Edition. http://www.zdnet.com/zdnn/stories/news/0,4586,2766234,00.html. May 2001
5. "The Gnutella Protocol Specification". Online. http://dss.clip2.com/GnutellaProtocol04.pdf. 2001
6. "[Gnutella] Network Improvements". Online. LimeWire Corp. http://www.limewire.com/index.jsp/net_improvement. March 2001
7. “Gnutella Network Snapshot”. Online. Gnutellameter.com. http://www.gnutellameter.com. 2001
8. Flinn and Maurer. “Levels of Anonymity”. Journal of Universal Computer Science. Volume 1. Number 1. Page 35-47. 1995