A BRIEF INTRODUCTION AND ANALYSIS OF THE GNUTELLA PROTOCOL

By
Gayatri Tribhuvan

University of Freiburg
Masters in Applied Computer Science

tribhuva@informatik.uni-freiburg.de

ABSTRACT

Peer to peer technology has certainly improved mechanisms of downloading files at a very high rate. Statistics and research show that as more number of peer to peer protocols developed, the number of nodes in the entire network increased.
In this paper, we observe the structure and operation of **Gnutella**, a peer to peer protocol. The paper also focuses on some issues of this protocol and what improvements were further made to improve its scalability.
We also have a look at some of the security issues of this protocol. Some statistics in the paper also reflect important changes that the protocol inflicted onto the entire network.

1. INTRODUCTION

Gnutella is a p2p protocol. This paper has tried to address some of the issues that have been faced by users of Gnutella, including that of scalability (increasing the scale of operation, i.e. the volume of operation with progressively larger number of users) and security.

The Gnutella protocol is an open, decentralized group membership and search protocol, mainly used for file searching and sharing.

Group membership is open and search protocol addresses searching and sharing of files.

The term Gnutella represents the entire group of computers which have Gnutella speaking applications loaded in them forming a virtual network.

Each node can function as both client as well as server. Thus they can issue queries to other nodes as well accept and respond to queries from other nodes, after matching the queries with the contents loaded in their own hard disks. Thus queries are not sent to any central server as in any client –server network, but are handled between the nodes.

Thus this network is unstructured and the distribution of files within this network can be totally random. However this poses a problem as number of nodes goes up. Since each node responds to a query from any node by sending a list of all its contents to the querying node, the load of information going to a querying node increases in direct proportion with the number of nodes in the network. This puts a limitation on the scalability (increasing the number of users in a network) of Gnutella.

2. DESIGN GOALS OF GNUTELLA

Like most P2P file sharing applications, Gnutella was designed to meet the following goals:

- Dynamic network which allows users to join and leave continuously. This goal has been achieved satisfactorily.

- Scalability – This as we have seen above poses a challenge. Considering the fact that this is a basic requirement in any p2p network, this is a serious problem.

- Reliability in the face of external attacks from viruses etc.

- Anonymity which is a basic privacy requirement in any p2p network

3. GNUTELLA PROTOCOL DESCRIPTION

3.1 Gnutella Protocol Features

- Types of communication descriptors: ping, pong, query, query hit and push.
- Incoming clients are assigned default values for each of these parameters.
- In order to assist newly entering clients to get familiar with the protocol and the type of parameters enjoyed by the older clients already in the network, client vendors have set up host-cache servers that provide the list of IP addresses of existing member nodes to any new entrant.

- Another way the above need is met is to provide the list of past IP addresses that have been members of the network. This way any new entry is instantaneously updated with this list and can then automatically connect with any of the past peer nodes immediately.
- Immediately after receiving the IP addresses, a client node may send several queries and receive query hit responses. However the actual sharing of the data through file transfer is done thru HTTP protocol, offline. This saves the Gnutella p2p protocol from handling the data load of file transfers.
- However if the file transfer is blocked by a fire wall then the Gnutella protocol may have to assist by deploying the push descriptor.

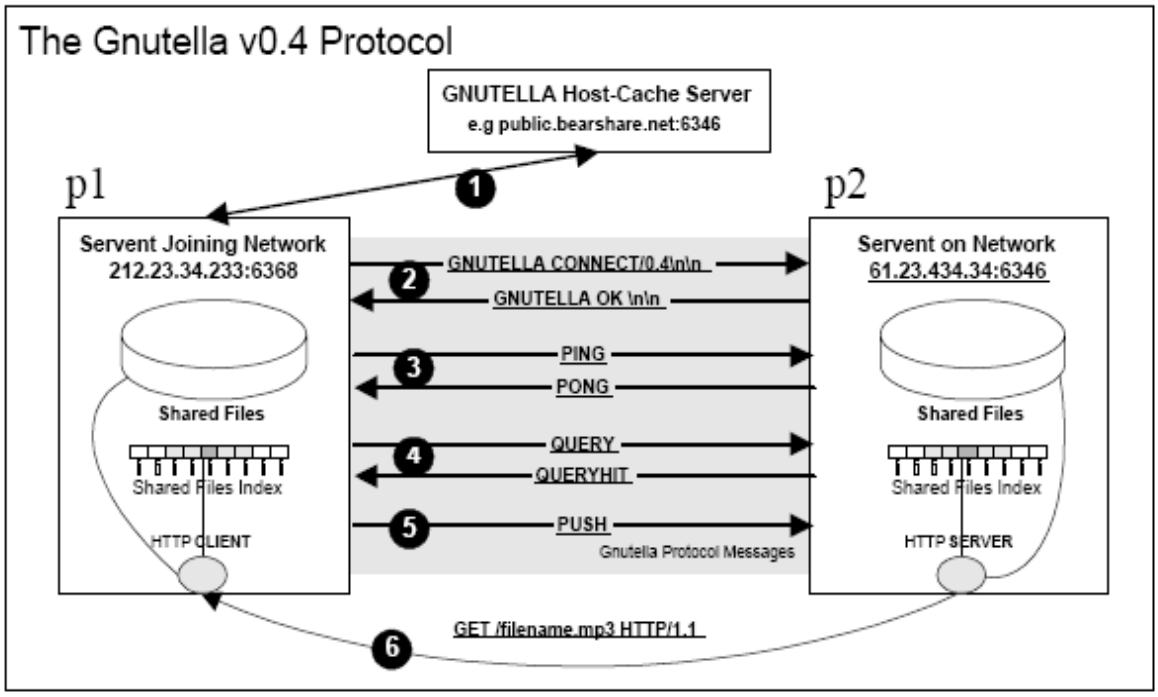


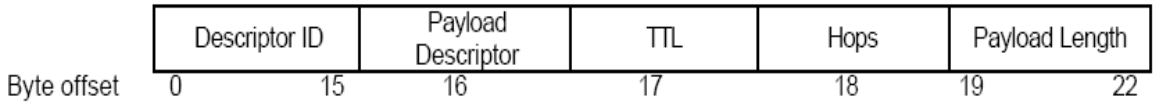
Fig. 1. Gnutella v0.4 Protocol.

Figure taken from [1]

3.2 Types of packets used for communication

Descriptor headers

These packets ensure successful connections of Servents to the Gnutella network. As soon as a prospective Gnutella participant joins the Gnutella network, it sends its Descriptor header to the other existing Gnutella participants.



Figures and descriptions in this section are taken from [3]

Ping

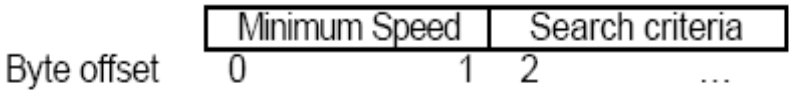
The ping query is often responded by the pong response. It is an optimal environment in the network if the ping traffic is minimised as much as possible.

Pong

Pong descriptors are responses to ping descriptors. It is possible to send more than one pong in reply to one ping.

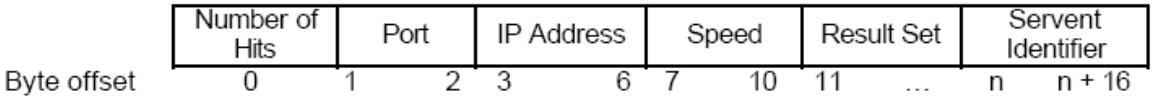
Query

Query is responsible for the search mechanism in the network. Minimum speed is required to match network requirements. Search criteria are the key factor of the search. It could be a string.

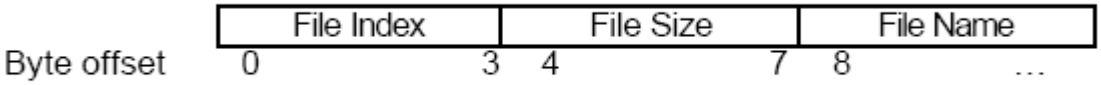


Query Hit

This is a response to a Query request.

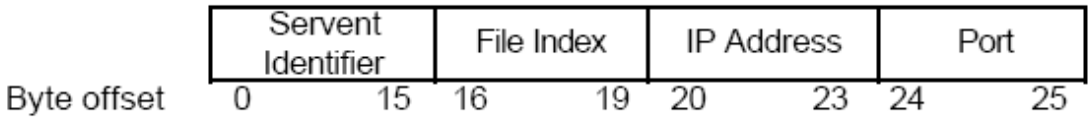


The results are stored in the result set and are compared with the given query. The result set looks like this



Push

Used when the servent containing the file is behind a firewall.



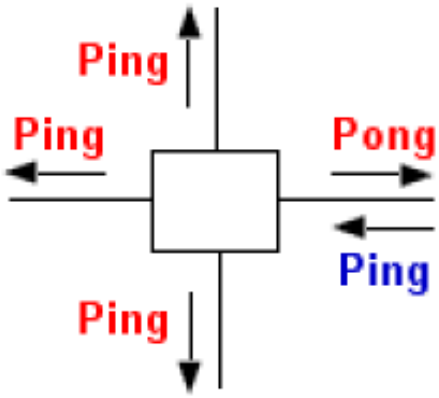
3.3 Descriptor routing

Figures and descriptions in this section from [3]

A well-behaved Gnutella servent will route protocol descriptors according to the following rules:

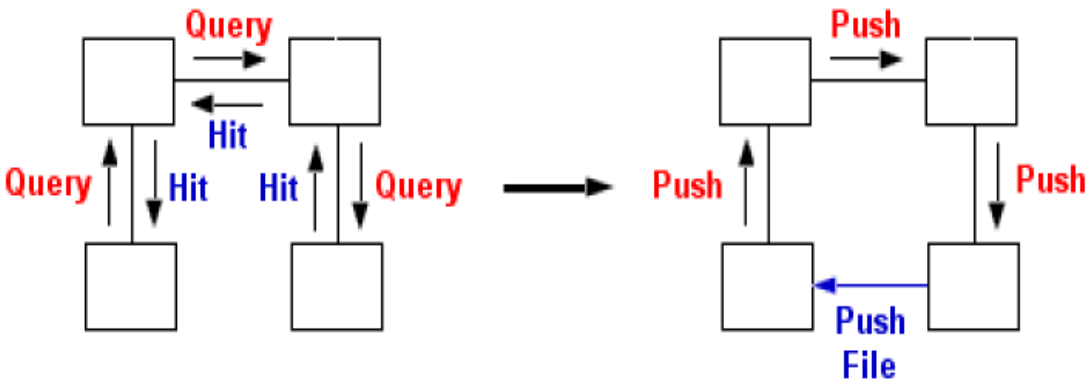
1. Ping Pong routing

Pong descriptors may only be sent along the same path that carried the incoming Ping descriptor. This is done so that just the servents that sent the Ping descriptor will see the Pong descriptor in reply. If a pong descriptor with a certain descriptor id n is generated, the ping with the same descriptor id is checked for. If it doesn't exist, the pong descriptor is removed from the network.



2. Query – Query hit- Push Routing

The same logic is followed in this routing. With respect to query and query hit routing, query takes the place of ping and query hit takes the place of pong. In case of Query hit and push routing, Query hits are pings and Push descriptors are pongs (in the routing rules).



4. MECHANISMS

4.1 Functioning Mechanism

How does a servent join the Gnutella network?

Direct connection on the network is established with the following command prompt:

GNUTELLA CONNECT/<protocol version string>\n\n

Where , <protocol version string> is defined to be the ASCII string “0.4”

Response is as follows:

GNUTELLA OK\n\n

Any other response indicates that servent is not agreeing to accept the connection. Rejection could be due to exhaustion of slots or many other reasons.

4.2 Searching Mechanism

Gnutella protocol uses the standard Breadth first search mechanism. The steps of search are as follows:

- Node initiates search for file
- Sends message to all neighbors
- Neighbors forward message
- Nodes that have the file initiate a reply message
- Query reply message is back-propagated
- File download occurs

- Note: if one client is behind a firewall, another servent can request that servent push the file to itself.

4.3 File Download

A direct connection is established between source and target for file download. File download always takes place out of the network and never on the network. Http protocol is used.

GET /get/<File Index>/<File Name>/ HTTP/1.0\r\n
Connection: Keep-Alive\r\n
Range: bytes=0-\r\n
User-Agent: Gnutella\r\n3\r\n

Protocol definition taken from [2]

File name and file index are corresponding entities from the query and query hit pairs.

5. DRAWBACKS OF GNUTELLA

- Since 50 % of files are served by just 1 % of the nodes, this becomes more like a client server network than p2p.
- Loss of network bandwidth and security through transfer of masquerading files.
- Problems of malicious peers.
- Distributed denial attack through spam. How far spam can travel in such cases is another question that needs to be resolved and curbed.
- PONG attacks
- Injecting viruses thru PUSH leaks
- Loss of anonymity thru GUID tracing

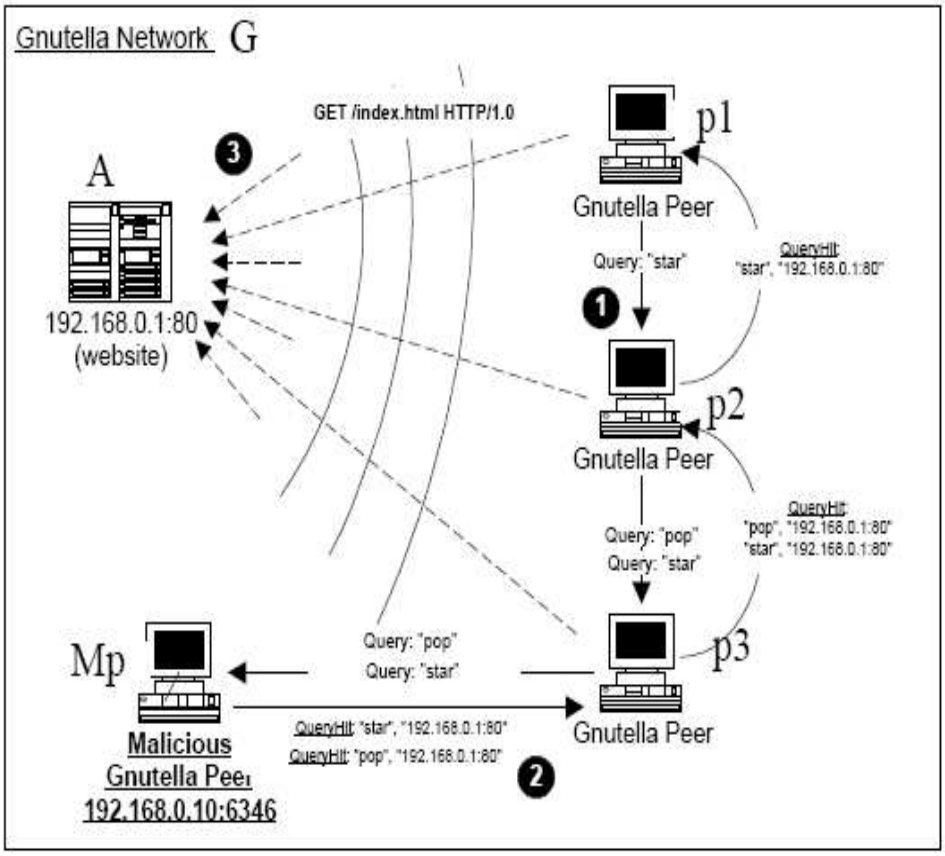
DETAILED CASES OF GNUTELLA WEAKNESSES

Distributed Denial Service

Here, malicious peers join the Gnutella network in addition to ordinary Gnutella clients. These malicious peers have intentions of harming other peers. They have attractive features like high bandwidth etc to attract as many peers as possible. Both

ordinary peers and malicious peers join the Gnutella network by exchange of ping and pong messages.

Figure taken from [1]

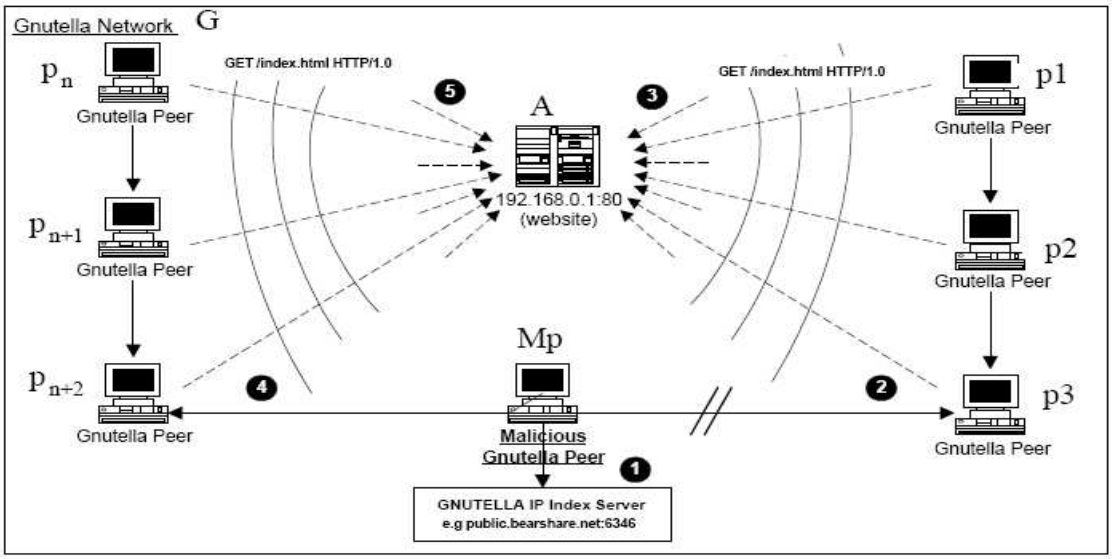


Performing a Distributed Denial Service on a public webserver (e.g. yahoo.com).

Distributed Denial Service attack through Spamming

Once a malicious peer becomes a part of the Gnutella network, it allows spammed messages to enter into the Gnutella network by replying to ping requests by prospective Gnutella members. In this manner these spammed messages travel through out the network.

Figure taken from [1]



How Far can these spammed messages travel?

This is a very difficult question to answer as there are number of factors to consider:

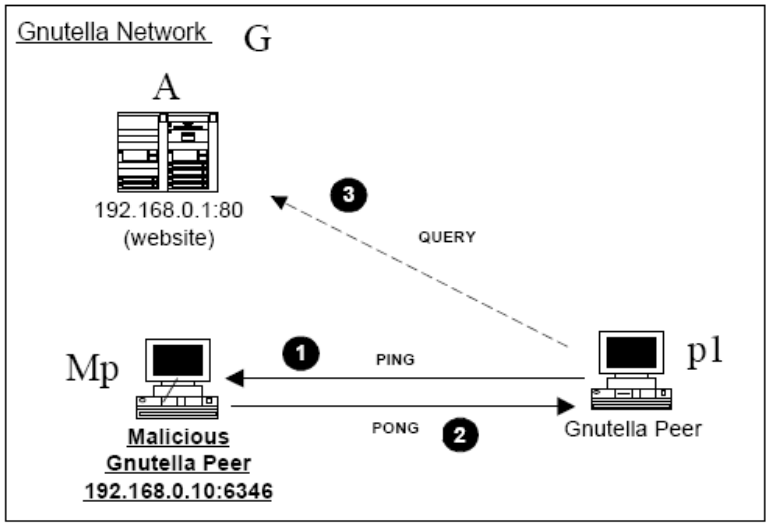
- number of peers that are connected to the Gnutella network
- the topology at the time of attack
- the number of connected hosts of the peer
- the maximum TTL of a peer
- the percentage of messages lost due to disconnections or failures.

Pong Attack

Since Mp is a misbehaving peer it will respond with a Pong which contains the IP and the PORT of the host in danger. p1,the ordinary peer would think that it established a connection with A, and forward all Query messages to A. But the good point is that, this attack will last only for a short period.

This happens because p1 after sometime sends a Ping message to all the hosts again, for route re-discovery. Since A is not really a Gnutella client and is not able to reply with a Pong message to the request, p1 will simply remove A and the attack will terminate.

Figure taken from the paper [1]



. p thinks that it is connected with A, instead of Mp, and forwards to A all *Query* Messages.

6. STATISTICS

Statistics of reliability of Gnutella networks

Power law distribution – appears as a line on a log-log plot.
Recent networks have moved away from the above distribution since there are too few nodes with adequate connectivity to form a power-law distribution. Such nodes follow quasi-constant distribution while the high connectivity nodes continue to follow the power-law distribution.
Uniform connectivity distribution helps cope with random node failures better. It also protects the network from attacks on highly connected nodes that are also easy to track and single out for attack.
Estimates of Traffic –
A crawler spy has unearthed the following distribution of traffic IN A GNUTELLA NETWORK –
QUERY – 92 %
PING – 8%
Average data traffic was 6 KBPS per node.
Total traffic for a large Gnutella network was estimated at 1 GBPS, for 170,000 nodes. This amounts to about 1.7% of total traffic in the USA. This shows that volume of traffic handled is a major obstacle for scalability as is also the efficient use of underlying network infrastructure

Graphs taken from [2]

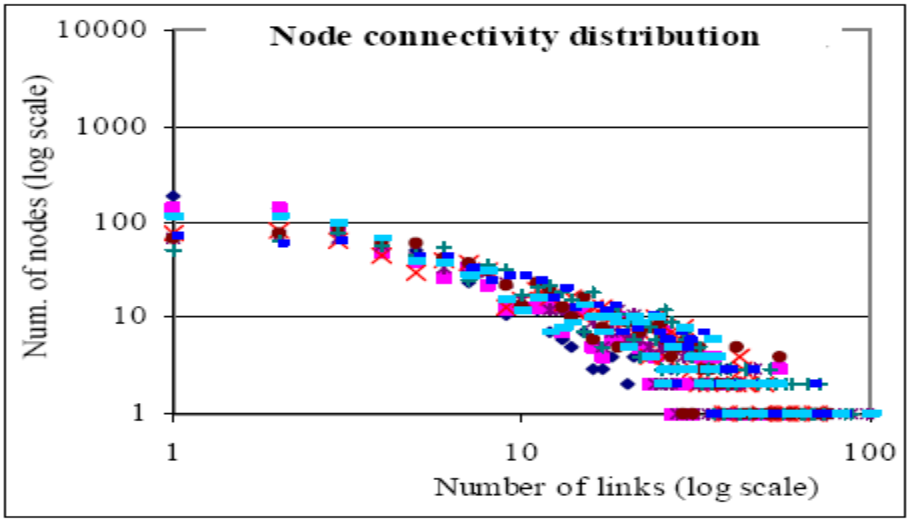


Figure 1: Connectivity distribution during November 2000. Each series of points represents one Gnutella network topology we discovered at different times during that month. Note the log scale on both axes. Gnutella nodes organized themselves into a power-law network.

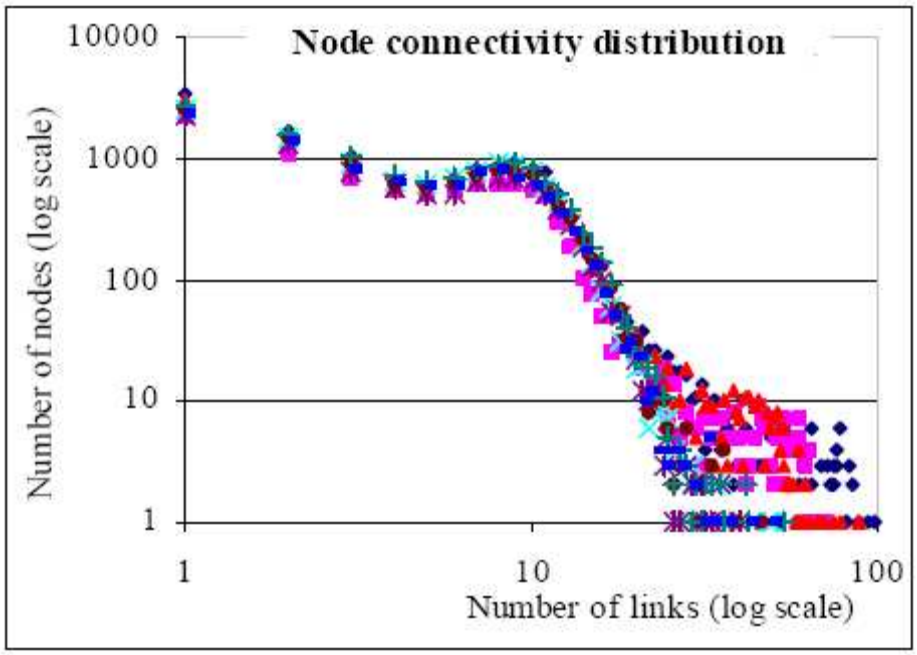


Figure 2: Connectivity distributions during March 2001. Each series of points represents one Gnutella network topology discovered during March 2001. Note the log scale on both axes. Networks crawled during May/June 2001 show a similar pattern.

7. GNUTELLA 2

How it works

Gnutella2 divides nodes into two groups: leaves and hubs. Leaves are connected to one or two hubs, and hubs can have many leaves and many connections to other hubs. When the search starts, the node gets a list of hubs required, and contacts these hubs in the list, checking which have been searched, until the list is exhausted, or until a search limit has been reached. So a popular file can be searched for without burdening the network.

Files of leaves are indexed by hubs by through query routing tables, which contain entries in the form of hashed keywords the leaf uploads to the hub. The hub combines this with all the hash tables its leaves have sent to send to its neighbors. By doing this bandwidth is reduced efficiently, as queries are not forwarded to leaves and neighboring hubs if the entries which match the search are not found in the routing tables.

Gnutella2 relies extensively on UDP, rather than TCP, for searches. The overhead TCP introduces would make a random walk search system unworkable, though UDP has its own disadvantages such as unreliable services, etc.

Protocol Features

- Gnutella2 has an extensible binary XML-like packet format
- Employs SHA-1 hashes for file identification and integrity check of files.
- Tiger tree hashes are used for downloading multiple instances.

In addition it uses a metadata system for file searching with special labels and input parameters. If a user wants one particular file, many files could exist with the same name, their content being different. This search solves this problem and finds only the required file the user wants.

Compression in the network is another feature.

Gnutella2 versus Gnutella

Both the networks are similar in most aspects, with major differences being in the packet format and the search methodology. Gnutella's packet format was not very efficient, so Gnutella2 learned from this, so besides having many of the added features of Gnutella standard in Gnutella2, extensibility was a key feature that was incorporated.

Another key difference is the search algorithm. While Gnutella used query flooding for searching of files, Gnutella2 uses a walk system where information is collected from a list of hubs and contacts are made directly, one at a time. This is better in terms of efficiency and less load. Queries are not routed through so many nodes and once enough information is obtained about the result, it allows the client to stop.

8. CONCLUSION AND FUTURE WORK

The Gnutella protocol basically has a decentralized characteristic and hence can perform better than P2P protocols like Napster. For instance, question of reliability cannot be answered in centralized P2P networks since these networks are solely dependent on their central access point, which if collapses due to some reason , disintegrates the entire network. In contrast, distributed models such as Gnutella have many access points and are more difficult to crumble down if some intermediate nodes are malfunctioning or disabled.
Gnutella node connectivity essentially combines a power law and a quasi-constant distribution. This property keeps the network as reliable as a pure power-law network and makes it harder to attack for a malicious entity.

More analysis is being done in the security aspects of Gnutella.

References

- Website: www.wikipedia.com

[1] Exploiting weaknesses of Gnutella by Demetrios Zeinalipour-Yazti

[2] ***Macroscopic Properties of Large-Scale Peer-to-Peer Systems*** ***Matei Ripeanu, Ian Foster***

[3] Gnutella protocol standard version 4