

GnuViz – Mapping the Gnutella Network to its Geographical Locations

Rüdiger Schollmeier, Gerald Kunzmann

Lehrstuhl für Kommunikationsnetze, Technische Universität München

Schollmeier@ei.tum.de, Kunzmann@ei.tum.de

Abstract

Gnutella is a classical Peer-to-Peer network designed for file-sharing. The absence of pure servers is one of its main properties, given that every Gnutella host is client and host in one. It uses the resources of the participants to distribute content, e.g. mp3 compressed audio files, and shares the processing capacity to provide the routing and searching capabilities for the network.

In this work we present GnuViz, a tool to visualize the geographical context of the virtual overlay network established by the Gnutella Protocol. Therefore a Gnutella Network Crawler is used, so that real life measurements of the Gnutella Network can be performed. With the additional aid of a geographical database the acquired IP address of each logged participant can be assigned to its geographical coordinates. A Java based script finally displays the network structure on arbitrary world maps. By using GnuViz we are able to substantiate the shortcomings of Gnutella, and propose protocol modifications to improve the network behavior of Gnutella.

1. Introduction

The traffic generated by Peer-to-Peer networks has been increasing steadily and has already become one of the major traffic factors in IP-based networks. Within the German research network [4] already 70% percent of the overall traffic is generated by Peer-to-Peer applications [6]. One of the most prominent applications within the area of Peer-to-Peer is Gnutella which is based on the Gnutella Protocol [17]. Gnutella is mainly used as a file-sharing application and establishes therefore a virtual overlay network built on TCP-IP connections to exchange service messages, like e.g. PING, PONG or QUERY messages, and to download files.

As Gnutella is based on a pure Peer-to-Peer network [14], most of the service messages are flooded over the networking, i.e. every incoming message is forwarded to all neighbours of the node which received the message. Thus a lot of bandwidth may be wasted, if the connections between the Gnutella nodes are established randomly. Although we were able to show in [13], that the scalability issues mentioned in [10] are solved by the network itself, we think that too much bandwidth is wasted, as the virtual network is not mapped properly to the physical network.

In its current specification, the log on to the network is not specified. However in the most implementations, a

Gnutella node first tries to contact a kind of beacon server, which stores a certain number of nodes, which recently already logged on to the Gnutella network, via this beacon server (hostcache) with a temporarily fix IP-address. Thus the beacon server is able to provide several IP-addresses of active participants.

Nodes wanting to participate thus first have to send a Gnutella PING-message to a beacon server. The beacon server answers this PING-message with several PONG messages, which include several IP-addresses of active Gnutella participants, to which the new node can now establish TCP-based Gnutella connections. Thus the new node becomes an active member of the Gnutella network. The second possibility to log on to the network is to use IP-addresses cached from previous Gnutella sessions cached IP-addresses. The node tries to establish a Gnutella connection to each of the cached IP-addresses, to find at least on node, which is still an active member of the network.

In both cases no preferences are used, to select the IP-addresses a new node establishes its connections to. As further on the Gnutella nodes are distributed nearly all over the world, the connections between the Gnutella nodes can also reach all over the world, and can also span the whole world.

Previous measurement approaches concentrated on traffic issues ([8], [5]) and the user behaviour [16] and not on the network structure. Other attempts to map the network only explore the network and do not map the network to its geographical locations [11]. They are thus not able to show the problems of Gnutella implied by the random assignment of connections.

With the help of GnuViz, we want to make this world spanning network visible. Thus we are able to show the problems caused by the random assignment of connections to Gnutella nodes. As an additional result we propose a solution to enhance the speed and message load of Gnutella.

Therefore we describe the conceptual architecture of the components of GnuViz in Section 2, i.e. the network crawler and the visualizing component of GnuViz. In Section 3 we present the results we are able to obtain with GnuViz and draw conclusions from the maps generated by GnuViz. Resulting from the shortcomings of Gnutella, detected by GnuViz and described in Section 2, we present an enhancement of the Gnutella protocol in Section 4, to avoid these shortcomings. This work is concluded in Section 0 by a discussion of the presented

work and a description of future research topics in the area of visualizing virtual network structures.

2. Conceptual Architecture of GnuViz

GnuViz consists of three major components. The first one is the network crawler, which explores the network structure, using the PING and PONG messages, as explained in detail in Section 2.1. The crawler stores the network structure, i.e. the IP-addresses of the nodes and the connections between these nodes in a table.

As soon as the network structure is analyzed to the desired extent, the location information retriever evaluates the geographical position of each explored node, as described in Section 2.2.

The third component then uses the location and the network information of the crawler and the location retriever, to draw the network on any desired geographical map and displays the map as a java-script enhanced HTML-document.

Although, GnuViz was developed for a Gnutella network, it can be used within other existing networks like Freenet [2] or JXTA [15] and future networks like in Chord [7], Pastry [1] or Tapestry [3] too. Due to its modular design, only the crawler has to be adapted.

2.1. The crawler

To explore the connections between the nodes participating in the Gnutella network, the PING and the PONG descriptors are used. Therefore a first starting point is needed, to be able to log onto the network. It is sensible to choose one of the available beacon servers as a starting point, as they are in most cases always available. The beacon server provides the crawler, with a certain number of active Gnutella nodes. The crawler tries to connect to the se nodes and is thus able to log on to the network.

As mentioned above Gnutella is based on TCP-IP connections. Therefore it is necessary, that the crawler first establishes a TCP-IP connection to one of the previously received IP-addresses, before being able to establish a Gnutella service connection to the assumed member of the Gnutella network. In the case of a successful setup of the TCP-IP connection, the Gnutella service connection is established by sending a Gnutella Connect message. The crawler can now record the node with its IP-address in its network table, as an active member of the Gnutella network. If the other node accepts additionally the Gnutella service connection setup, the Gnutella service connection is automatically established. Thus the crawler is now ready to explore the network beyond the node, to which the crawler is now connected. In the case, that no Gnutella service connection can be established, no further effort is made to investigate the network beyond this node, as it is not reachable for us.

To explore the network beyond an active and reachable node, the crawler sends a PING message on this connection with the Time-To-Live value set to two hops. Due to the TTL value set to two, the Ping message is forwarded only to the neighbors of the currently considered Gnutella node, according to the Gnutella protocol. Every Gnutella node which receives a PING message announces its own presences and in some implementations, also the presence of its neighbors, by sending back one or several PONG-messages to the originator of the PING message. Every PONG message includes besides other information, the IP-address of the answering node and the number of shared files. The IP-address received with a Pong message is recorded in the network table together with the parameters of its Gnutella connections. Additionally the IP-address is put into a list of nodes beyond which the network has to be analyzed further in the next hop level. The additionally information transmitted in a PONG message, i.e. the number of shared files is stored in a second list for further statistical analysis.

The above described procedure is repeated iteratively for all nodes in the same hop distance, before proceeding to the nodes stored in the next hop level list. Thus the network crawler explores the whole Gnutella network up to any user configured hop level. The result is an interlinked tree-like structure, with the beacon server as the root node and the explored Gnutella nodes as its branches.

To avoid dead ends and not to get stuck in not responding nodes, a timer observes every active TCP-IP connection and closes it, after a predefined time, regardless whether a Gnutella connection is active, or whether the crawler still tries to connect to a node. A value of 5ms proved to be suitable during our crawls.

As the Gnutella network is a very dynamic network, it is necessary to explore the network as fast as possible. Therefore the crawler is able to explore the network on several connections in parallel, depending on transfer rate and processing restrictions. Thus it is possible to crawl the network faster and to retrieve more accurate network information.

2.2. The location retriever

The location retriever maps the IP addresses of the explored nodes to their geographical locations, i.e. their longitude and their latitude. The basis for our location retriever is NetGeo [12]. Netgeo, developed within the CAIDA project, is an Internet geographic database and collection of perl scripts, which can be used to map IP-addresses, domain names and autonomous system numbers to their geographical location.

To find out the latitude/longitude values for a domain name, NetGeo first searches for a record containing the target name in its own database. The NetGeo database

caches the location information parsed from the results of previous who-is lookups, to minimize the load on who-is servers. If a record for the target domain name is found in the database, NetGeo returns the requested information, e.g., latitude and longitude. If no matching record is found in the NetGeo database, NetGeo performs one or more who-is lookups using the InterNIC and/or RIPE whois servers, until a whois record for the target domain name is found.

After obtaining a record from a who-is server, the NetGeo Perl scripts parse the who-is record and extract location information and the date of last update. The NetGeo parser attempts to extract the city, state (or province, district, etc.), and country from the text of the who-is record. For US addresses the parser also extracts the zip code, if possible. If the parser is unable to parse an address it attempts to find an area code or international phone code in the contact section; the phone code is mapped to a country and then the parser attempts to parse the address again, using the hint provided by the phone code. The parser also guesses the country from email addresses with 2-letter TLDs found in the contact section.

There are several other geographical databases existing, but none of them are reliable enough. Unlike NetGeo, they either return only the country, respective the geographical position of its capital, or they only have a

very limited number of records. Even a combination of different databases proved to be far from sufficient.

To automate the mapping of the IP-addresses to their geographical location, we use a CGI interface, offered by the NetGeo server. The server rate-limits access to 30 lookups in 30 seconds from a given source IP address. However, as we look up the location of the IP-addresses offline, this restriction is not a real problem. The size of the crawls is mostly about some thousand IP addresses, so they can be assigned to their geographical information within a few hours, as 3600 IP-addresses can be looked up within one hour.

During the mapping process, time is not a critical factor anymore, as all possibly varying objects have already been captured in the snapshot of the Crawl.

A future development of the program could include a real time map of the Gnutella Network. Then a higher rate is inevitable, but for this purpose CAIDA's NetGeo offers a cooperation with research institutes.

As soon as the topological information for every IP-address of any member of the Gnutella network is retrieved from the NetGeo-Server, the maps as displayed in Figure 1 can be created. Therefore the nodes are marked on the world map, according to their longitude and latitude and the explored connections between the nodes are added at last, resulting in a map as presented in Figure 1.

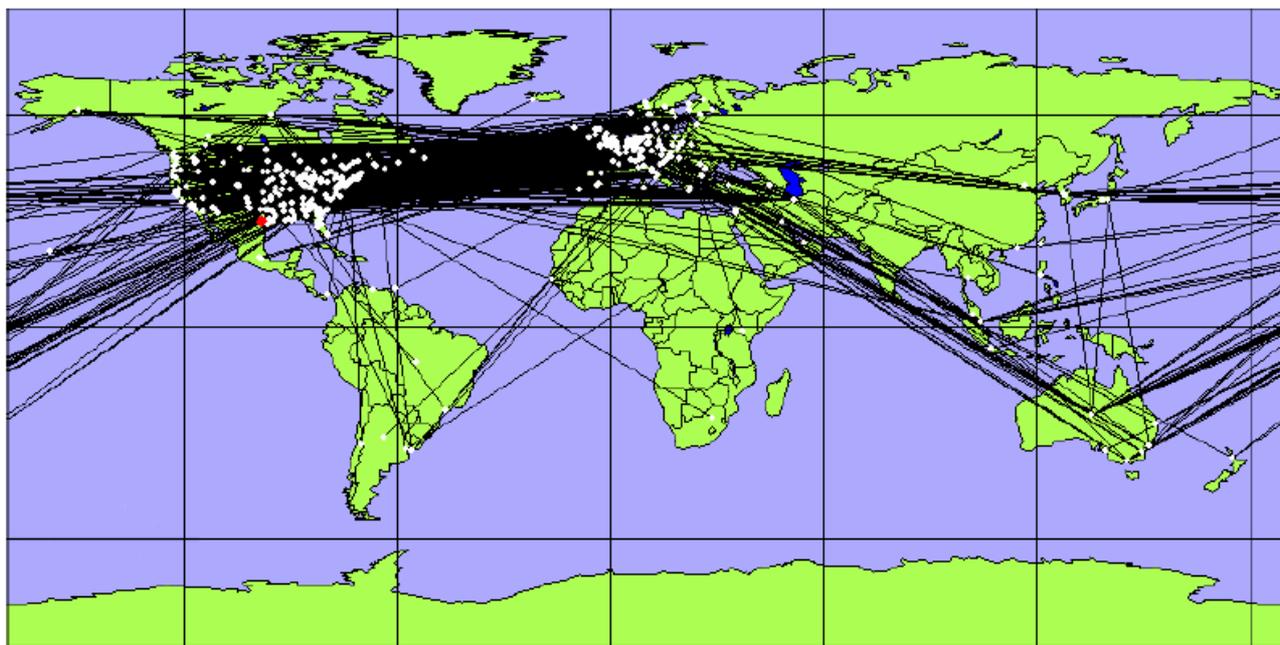


Figure 1 Connections Between the Gnutella Nodes (01.08.02)

3. Results

As it can be seen from Figure 1 the Gnutella nodes are not evenly distributed over the world. The Gnutella nodes are mainly concentrated in the USA and in Europe, which matches with the distribution of Internet users worldwide.

Table 1 Distribution of Internet Users Worldwide
[www.nua.ie/surveys/how_many_online]

World Total	580.78 million
Africa	6.31 million
Asia/Pacific	167.86 million
Europe	185.83 million
Middle East	5.12 million
Canada & USA	182.67 million
Latin America	32.99 million

A remarkable detail of Figure 1 is the broad black bar between Europe and the USA. This results from the high amount of Gnutella connections between nodes in the USA and nodes based in Europe. More than 32% of all connections between Gnutella nodes are established across the Atlantic, i.e. between Europe and the USA. Further on the black lines between the Gnutella nodes represent only virtual connections based on TCP-IP. Thus it must be assumed, that even more connections established between Europe and Australia or Asia are routed via the USA, and thus put additional traffic on the Internet connections between Europe and the USA.

As in the DFN already 70% of the overall traffic is generated by Peer-to-Peer applications, this high amount of data should lead to an increasing symmetry between the USA and Europe, as Gnutella traffic itself is symmetric.

The high number of long Gnutella connections, especially across the Atlantic is a real drawback for the Gnutella network. From our point of view it is a huge waste of bandwidth, to search for objects not only in the nodes local area. We expect the probability to find a certain content in the geographically surrounding nodes to be remarkably higher, as the interests within one geographically close area are expected to be much more similar.

The above mentioned random assignment of new nodes to active nodes is an additional shortcoming of the current Gnutella architecture. In Figure 2, Figure 3 and Figure 4 the hardly acceptable consequences of the random network establishment are highlighted. In Figure 4 the zigzag structure of the network after 4 hops can be observed clearly. Figure 2 and Figure 3 show the development of the zigzag structure. The route starts in New Mexico/USA, its first hop leads across the Atlantic to Sweden, the second back again to California and it finally terminates after the third hop in Germany. Thus any message routed through

the node in Germany or the node in New Mexico, crosses the Atlantic ocean at least three times before reaching its destination.

These zigzag routes, which are common within the Gnutella network, lead to a high but unnecessary consumption of bandwidth between Europe and the USA. Further on higher delays have to be taken into account, as the messages have to take long routes, to find certain content, which most probably is available at geographically closer located nodes. It would be much more sensible to route the message only once across such long distances and then to distribute it, if necessary within the respective local area.

Based on our experiences we assume, that more interests, especially music interests, are shared within a local area, than in fares away sites. Therefore it would be more efficient for the user, as well as for the network, to keep the traffic in a certain geographically near area. Thus it would be sensible to form clusters, which comprise of several geographically closely located nodes. The search would then start first in these clusters. If the content can not be found in these clusters, the search is extended to nodes which are farer away. In the following chapter we therefore propose an extension to the Gnutella protocol, to offer the possibility to form these clusters.

A further result we obtained with GnuViz is that we can prove the existence of several independent networks within the Gnutella network. We were never able to explore all nodes, as well as we believe according to [9], that there have to be more. We seldom explored nodes in Canada or Japan independent of the crawling depth. We thus assume, that nodes belong to different networks. We just did not enter the network at the right place. In our further research we therefore want to explore these networks too, by choosing different starting points for our crawls.



Figure 2 Map of the Gnutella Network on 01.08.02, up to Hop Level 1



Figure 3 Map of the Gnutella Network on 01.08.02, up to Hop Level 2

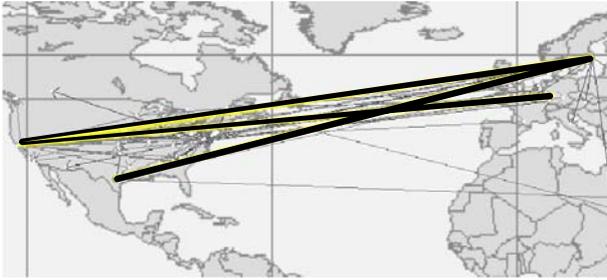


Figure 4 Zigzag PING-PONG-Route (01.08.02)

4. Gnutella Enhancements

The high consumption of bandwidth is one of the most often criticized characteristics of Gnutella. Therefore problems, like the development of zigzag routes as described above, should be avoided and the automatic development of clusters based on the location of each node should be supported by the network. The motivation therefore is, that we think, that more interests are shared in a geographical proximity, and that thus a higher rate of queries can be accomplished, because of the shorter delays.

As a result of our observations we propose two extensions of the Gnutella descriptors PING and PONG. Thus we want to provide automatic geographical clustering. We propose an extension of the two descriptors by 8 bytes and only change the internal message handling and not the overall routing of the messages. The descriptors are each extended by two fields, which both keep a float value, describing the longitude and the latitude of the node, which sent out the respective message. These extensions would not be necessary, if it would be possible to derive the geographical location of a node from its IP-address. Unfortunately this is not possible as the IP-address range and its sub networks is distributed mostly independent of its geographical location all over the world.

On receiving a PING message with a valid latitude and longitude field, the beacon server should answer an initial PING not with a simple FIFO-list of active nodes, but with a geographically filtered FIFO list of currently active nodes. It should only return an ordered list of IP-addresses of those nodes. This list is ordered by the geographically closeness to the requesting node. Thus this node can enter from the start a cluster of geographically near nodes.

The internal message handling of the normal nodes is organized in a similar way. With the help of PONG descriptors, received IP-addresses of active nodes are cached in a ordered list. This list is sorted according to the geographical distance of the node.

In the case, that a node wants to establish a new connection, it tries to establish a connection to the nearest available node, by using the above described list. Further on a location aware Gnutella node should replace existing connections if it receives an address of a node which is closer than a node it is currently connected to. Thus

automatically clusters evolve which have similar interests and a similar location in common. This concept also works in conjunction with other enhancements of the Gnutella protocol, e.g. the PING-PONG schemes or the Ultrapeer-concept. Our proposal does not change the routing schemes of Gnutella. Even with Ultrapeers it would be much better, if a Ultrapeer could also be chosen according to its geographical position.

Table 2 Extension of the PING-Descriptor

Byte	Description	Value
0-15	Message ID (GUID)	
16	Payloadtype	0x00
17	TTL	
18	Hops	
19-22	Payload Length	0x08
23-26	Longitude (Float)	
27-30	Latitude (Float)	

Table 3 Extension of the PONG-Descriptor

Byte	Description	Value
0-15	Message ID (GUID)	
16	Payloadtype	0x01
17	TTL	
18	Hops	
19-22	Payload Length	0x18
23-24	Portnumber	
25-28	IP-address	
29-32	Nr. of shared files	
33-36	Kilobytes shared	
37-40	Longitude (Float)	
41-44	Latitude (Float)	

To get the geographical position of a node into the Gnutella application we think of three possibilities. First of all it should always be possible to let the user manually configure the settings. This may on the one hand overburden the user, but on the user it offers the user the possibility to jump directly into other clusters if necessary. Further on it may be possible to retrieve the information directly from the device, if the device is already location aware, due to a GPS-module, or any other location sensitive module. The third option we propose is a self configuring software-unit, which retrieves its position with the help of NetGeo or any similar who-is-server. This would only result in one small inquiry at the beginning of each session, which could be conducted by a small software module, which can be completely independent of the Gnutella application.

Such an extension would be completely downward compatible, and could be added to any new client, with a

comparable small expenditure. The Gnutella protocol and its routing schemes is kept completely the same, just the internal message handling and the messages are changed slightly. No additional hardware is necessary, as this protocol extension simply demands either more user interaction or an additional inquiry to the NetGeo server.

With these extensions of the Gnutella protocol, a much better adaptation to the physical network can be achieved. This results in shorter delays, and a significant reduction of the resource consumption, especially according to the transfer rate. Zigzag routes can thus be avoided, which results in shorter delays. The content may be found faster, as the search is started in the local surrounding, where more interests are shared.

5. Conclusion

With the help of GnuViz it is possible to get an exact snapshot of the Gnutella network, whose nodes are also mapped to their geographical locations. Thus the distribution of Gnutella participants on the different countries can easily be retrieved and their interconnections can be made visible. As a result we can show with GnuViz, that the virtual Peer-to-Peer network is not matched at all to the underlying physical network. This leads to zigzag routes across the Atlantic, which results in a significant waste of bandwidth and high delays between PINGS and corresponding PONGS.

To overcome these shortcomings we propose an extension of the PING and PONG descriptors by 8 bytes, so that these descriptors include the longitude and latitude of the node which generated the descriptors. By adding such a location awareness to the Gnutella network, the network structure can be improved significantly, and its message load and bandwidth consumption can be decreased significantly, as the virtual network is now mapped much better to the physical network.

The location awareness can be achieved by small extensions of the two descriptors. No additional interferences to the protocol, especially into its routing schemes are necessary. To make the location awareness effective it is just necessary to change the internal message handling of the Gnutella nodes, i.e. of the Beacon Servers, of the Ultrapeers and the normal nodes. Thus interworking with other non location aware nodes can be guaranteed.

Further developments in this area are to make GnuViz faster, so that the network can be explored faster, and thus that to make it possible to draw a life map of the network, with nodes leaving and entering the network and changing their connections.

In the area of context aware clustering, we want to cluster the participants of a Gnutella like network depending on a whole profile, i.e. their location, their interests, their possibilities or their business profile.

6. References

- [1] A.I.T Rowstron, P.Druschl. "Pastry: Scalable, Decentralized Object Location and Routing for Large-Scale Peer-to-Peer Systems". Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001). November 2001.
- [2] A. Langley. "Freenet". Appeared in "Peer-to-Peer Harnessing the Power of Disruptive Computing". Editor: A. Oram. Publisher: O'Reilly & Associates, Incorporated. March 2001
- [3] B. Zhao, J. Kubiatowicz, and A. Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, Computer Science Division, U. C. Berkeley, April 2001.
- [4] Deutsches Forschungsnetz.www.dfn.de. October 2002
- [5] E. P. Markatos. "Tracing a large-scale Peer-to-Peer system: an hour in the lifetime of Gnutella". 2nd IEEE/ACM Int. Symposium on Cluster Computing and The Grid. 2002
- [6] G. Foest, R. Paffrath. "Peer-to-Peer (P2P) and beyond". DFN Mitteilungen 58-3. 2002
- [7] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. "Chord: A scalable peer-to-peer lookup service for Internet applications". Technical Report TR-819, MIT, March 2001.
- [8] K. Anderson. "Analysis of the Traffic on the Gnutella Network". www-cse.ucsd.edu/classes/wi01/cse222/projects/reports/. November 2001
- [9] Limewire: Current Network Size. http://www.limewire.com/historical_size.html. January 2002
- [10] M. J. Jovanovic, F. S. Annexstein, K. A. Berman. "Scalability Issues in Large Peer-to-Peer Networks – A Case Study of Gnutella" <http://www.ececs.uc.edu/~berman/gnat/> November 2001
- [11] M. Ripeanu, I. Foster, and A. Iamnitchi. "Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design". IEEE Internet Computing Journal, 6(1), 2002.
- [12] NetGeo - The Internet Geographic Database. www.caida.org/tools/utilities/netgeo/. October 2002
- [13] R. Schollmeier, G. Schollmeier. "Why Peer-to-Peer (P2P) Does Scale: An Analysis of Peer-to-Peer Traffic Patterns". IEEE International Conference on P2P Computing 2002, Linköping Sweden. September 2002.
- [14] R. Schollmeier. "A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications". IEEE Conference P2P 2001, Linköping Sweden. August 2001.
- [15] S. Botros, S. Waterhouse. „Search in JXTA and Other Distributed Networks“. Proceedings of the 1st IEEE International Conference on Peer-to-Peer Computing (P2P 2001). Linköping, Sweden. August 2001
- [16] S. Saroiu, P. K. Gummadi, S. D. Gribble. "A Measurement Study of Peer-to-Peer File Sharing Systems" Technical Report # UW-CSE-01-06-02. July 2001
- [17] T. Klingberg, R.Manfredi. "Gnutella 0.6 RFC". June 2002, rfc-gnutella.sourceforge.net/draft.txt
- [18] V. N. Padmanabhan, L. Subramanian. "An Investigation of Geographic Mapping Techniques for Internet Hosts". ACM SIGCOMM'01. San Diego, USA. August 2001