An Efficient Topology Formation for Gnutella Network

Santosh Kumar Shaw

An Efficient Topology Formation for Gnutella Network

Synopsis of the Thesis to be submitted in Partial Fulfillment
of the Requirements for the Award of the Degree of

MASTER OF TECHNOLOGY

by

Santosh Kumar Shaw

Under the supervision of

Prof. Niloy Ganguly

Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
West Bengal, INDIA 721302

May 2008

Synopsis

1. Introduction

A peer-to-peer (P2P) system is a network that relies on the computing resources available from all the participants in the network instead of concentrating on a small number of centralized servers. These networks are overlay networks, useful for many purposes like file-sharing, streaming media, internet telephony, distributed computation etc. Depending upon the topology formation, P2P networks are broadly classified as structured and unstructured. Structured P2P networks ensure that any peer can efficiently route a search to some peer that has the desired files, even if the file is extremely rare. In these networks, the placement of resources are very controlled and defined, which arise the necessity of a structured pattern in overlay links. An unstructured P2P network is formed when the overlay links are established arbitrarily and invites many challenges.

Decentralized (fully distributed control), unstructured P2P networks (Gnutella, FastTrack, etc) are the most popular file-sharing overlay networks. The absence of a structure and central control makes such systems much more robust and highly self-healing compared to structured systems [7, 10]. But the main problem of these kinds of networks is scalability due to generation of large number of redundant messages during query search. Consequently as these networks are becoming more popular the quality of service is degrading rapidly [4, 8].

To make the network scalable, *Gnutella* [1, 2, 3] is continuously upgrading it's features and introducing new concepts. All these improvements can be categorized into two broad areas: improvements of search techniques and modification of the topological structure of the overlay network to enhance search efficiency. In enhanced search techniques, several improvements like *Time-To-Live* (TTL), *Dynamic query*, *Query-caching* and *Query Routing Protocol* (QRP) have been introduced. One of the most significant topological modifications in unstructured network was done by inducing the concept of *super-peer* (ultra-peer) with a two-tier network topology.

An introduction of Gnutella Network: The basic Gnutella consists of a large collection of nodes that are assigned unique identifiers and which communicate through message exchanges. Current Gnutella is a two-tier overlay network, consisting of two types of nodes : *ultra-peer* and *leaf-peer*. An ultra-peer is connected with a limited number of other ultra-peers and leaf-peers. A leaf-peer is connected with some ultra-peers. However, there is no direct connection between any two leaf-peers in the overlay network. The network follows limited flood based query search in which a query is propagated up to a predefined number of hops (TTL) from the source peer. Gnutella 0.6 incorporates *dynamic querying* over limited flooding as query search technique. In dynamic querying, an ultra-peer incrementally forwards a query in 3 steps ($TTL(1)$, $TTL(2)$, $TTL(3)$ respectively) through each connection while measuring the responsiveness to that query. Modern Gnutella protocol uses *QRP* technique over dynamic querying in which a leaf-peer creates a hash table of all the files it is sharing and sends that table to all the immediate ultra-neighbors. As a result, when a query reaches to an ultra-peer it is forwarded to only those connected leaf-peers which would have query hits [1, 2]. In the Gnutella network, handshake protocol is used to make new connections. A handshake consists of 3 groups of headers [1, 2]. The steps of handshaking is elaborated next:

1. The program (peer) that initiates the connection sends the first group of headers, which tells the remote program about its features and the status to imply the type of neighbor (leaf or ultra) it wants to be.

2. The program that receives the connection responds with a second group of headers which essentially conveys the message whether it agrees to the initiator's proposal or not.

3. Finally, the initiator sends a third group of header to confirm and establish the connection.

This basic handshake protocol is modified to overcome the problem of message overhead in our work.

Related Work: Many algorithms exist in the literature which modify the topology in unstructured P2P networks to solve the excessive traffic problem and improve query hits by using replication techniques. The structural mismatching between the overlay and underlying network topology is alleviated by using location aware topology matching algorithms [5, 6]. Papadakis *et* al. presented an algorithm to monitor the ratio of duplicated message through each network connection and the node does not forward any query through that connection whose ratio exceeds certain threshold [9]. Zhu *et* al. very recently presented a distributed algorithm in [13] to improve the scalability of Gnutella like networks by reducing redundant messages. They have pointed the same concept of elimination of 3 and 4-length cycles. However this is demand driven and involves a lot of control overhead. Also it is not clear how the algorithm will perform in the face of heavy traffic. The algorithm also does not take care in preserving the Gnutella parameters (like degree distribution, average peer distance, diameter, etc), hence robustness of the evolved network is not maintained. In our work we take into considerations all the above aspects and propose a holistic approach to topology formation. The algorithm initiates as soon as a peer enters in the network rather than having it demand driven. Various alternatives of data replication, search techniques and network topology for Gnutella network discussed in [8].

2. Contributions

The main goal of our work is to improve the scalability of the Gnutella network by reducing redundant messages and increase the number of query hits (results). One of the ways to achieve this is to modify the overlay network, so that small size loops get eliminated from the overlay topology. We make five contributions : (1) We propose a class of completely distributed handshake protocols (HPC-r) which generates a cycle-r network (a network which does not have any cycles up to length (r − 1) is referred as cycle-r network). (2) We show that our approach can be deployed into the existing Gnutella network without disturbing any of its parameters. (3) Through simulation results we shown that cycle-r networks are very effective for Gnutella's dynamic query search over limited flooding. Structural analysis indicates that the proposed network is as robust as existing Gnutella network. (4) We propose and evaluate a new index table replica placement technique for our networks to improve query hits and reduce both latency and search cost. (5) We analyze some design issues related to hit ratio of the network.

3. Evaluation by Simulation

To validate our approach, we have performed numerous experiments. We have taken different sizes (up to 1000k nodes) of networks and performed experiments on those networks several times to obtain the average behavior. Through these experiments, we have shown that HPC-r performs better than the existing protocols. In this section we have analyzed search performance, robustness, hit ratio based design and replication techniques of cycle-5 and cycle-6 networks through simulation. By these analysis we will get an idea of study the performance of cycle-r (r > 6) networks.

Simulated Gnutella: To generate existing Gnutella network, we have simulated a strip down version of Gnutella 0.6 protocols which follows parameters of

Limewire [1]. Our simulated Gnutella network exhibits all features (like degree distribution, diameter, average path length between two peers, proportion of ultra-peers, etc.) exhibited by Gnutella network. These features are obtained from the snapshots collected by crawlers [3, 11, 12]. The properties of the Gnutella and simulated Gnutella networks are given in the table 1 [1, 12].

Table 1: Properties of Gnutella and Simulated Gnutella Network

Property	Gnutella	Simulated Gnutella
No. of Peers	2000k	100k
Ultra-peer ratio	15-16% of total peers	15% of total peers
Avg. Diam. of ultra-layer	6-7	4-5
Maximum connections	ultra-ultra 32 ultra-leaf 30 leaf-ultra 3 (Applicable to Limewire)	ultra-ultra 32 ultra-leaf 30 leaf-ultra 3
Average Connections	d_{uu}: 25-26 d_{ul}: 20-22 d_{lu}: 3-4	d_{uu}: 22-23 d_{ul}: 17-18 d_{lu}: 3

Search Performance: Network coverage (number of unique peers explored during query propagation) and message complexity (average number of messages required to discover a peer) are taken as search performance metrics. The summery of search performance (relative to Gnutella network) is shown in the table 2 which reflects that cycle-5 and cycle-6 networks are better than Gnutella network in terms of both message complexity and network coverage. Cycle-6 networks show slightly better performance than cycle-5 networks in TTL(3). Gnutella network uses dynamic querying over limited flooding and uses $TTL(3)$ rarely for rare searches only. Also to maintain cycle-6 networks (HPC-6) required more bandwidth. So cycle-5 networks are good enough in performance to save bandwidth. For that we mainly focus on cycle-5 networks.

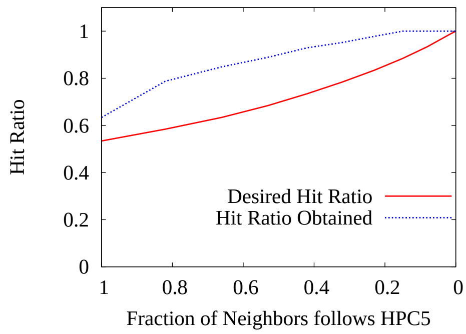
Table 2: Simulation Results for Search Performance and Robustness

		Cycle-5	Cycle-6
Search Perf.	Network Coverage (TTL-2)	15-20% more	15-20% more
	Network Coverage (TTL-3)	15-20% more	15-20% more
	Message Complexity (TTL-2)	10-15% less	10-15% less
	Message Complexity (TTL-3)	6-7% less	almost 20% less
Robustness	Random Removal	80-85%	—
	pathologically Removal	6-7%	—

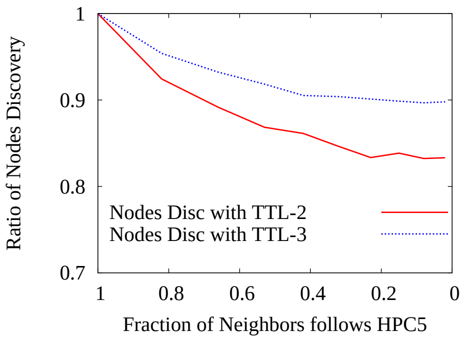
Comparison of Robustness: To test robustness of the network, we removed peers from the network in two ways: (i) random removal and (ii) pathologically removal (the highest-degree nodes first) [12] and find the largest component. In simulation , we have seen that the network gets fragmented after removal of certain percentage of peers (shown in table 2) which is similar to Gnutella network.

Hit Ratio Based Design: Hit ratio (inverse of the number of trials required to get a valid ultra-peer neighbor) is one of the main hurdle in the implementation of our approach. The problem of hit ratio can be solved by providing some level of flexibility in the neighbor selection. A peer may be allowed to select some neighbors in spite of creating smaller length cycles in the network. In simulation we have restricted a minimum fraction (say, r_x) of neighbors of each peer to be selected through HPC-5. In the Fig. 1 we have plotted the change of hit ratio and relative performance of the networks (with respect to the performance of cycle-5 networks) with different r_x. A system designer sets the hit ratio depending on the requirement of the networks/applications.

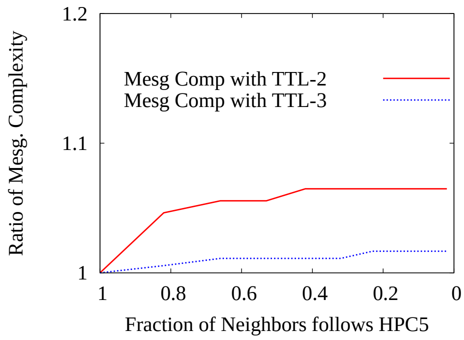
Replica Placement Techniques: We increase query hit probability (QHP) with less latency and less message complexity using replication techniques in cycle-5 networks. We have evaluated mainly two types of replica placement tech-



(a) Hit Ratio



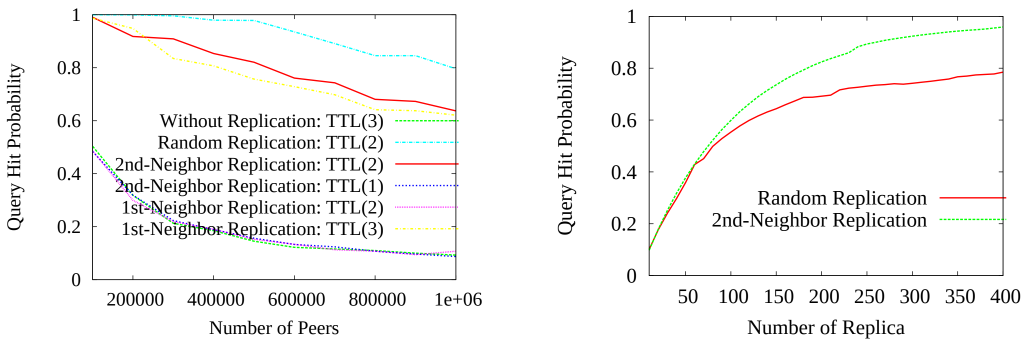
(b) Network Coverage



(c) Message Complexity

Figure 1: Effects of r_x in the network

niques (RPT): random RPT (replica is placed in randomly selected peers) and 2^{nd}-neighbor RPT (replica is placed in a subset of 2^{nd}-neighbor set). In the Fig. 2(a) have shown the QHP without and with different RPTs. Although in simulation, random RPT is better than 2^{nd}-neighbor RPT, we prefer to follow later one because of the limitation in uniform random ultra-peer selection in Gnutella network without having global knowledge. In Fig. 2(b) we have plotted QHP based on number of replicas of each MIT for both random and 2^{nd}-neighbor RPT. The QHP increases very rapidly at the initial stages and reaches almost at 0.8 in random RPT (0.7 in 2^{nd}-neighbor RPT) with 200 number (which is almost $0.5 * (d_{uu}^2)$) of replicas.



(a) Comparison of RPTs

(b) QIP Based on No. of Replicas with TTL(2)

Figure 2: Query Hit probability

4. Conclusion and Future Works

In this paper, we have presented a handshake protocol which is compatible with Gnutella like unstructured two-tier overlay topology. We have shown that the protocol is far more efficient than existing protocols. A relation among TTL, minimum cycle length in the topology and network performance has been observed and proposed. We have also presented a replica placement technique over our topology to increase the query hits which is very effective in Gnutella-like networks for very rare searches.

A major fraction of internet bandwidth is occupied by Gnutella-like unstructured popular networks. P2P implementation of the 2^{nd} generation web applications requires a huge internet bandwidth which initiates the optimum utilization of bandwidth. In this regard our protocol can be instrumental in improving the scalability and data availability of P2P networks.

In our future work we want to model our approach theoretically and planning to investigate on some of the issues (like stability, load balancing in replication) of our topological structure.

Bibliography

[1] Gnutella and Limewire: www.limewire.org.

[2] The Gnutella Protocol Specification 0.6: http://rfc-gnutella.sourceforge.net.

[3] Gnutella: www.gnutellaforums.com.

[4] Karbhari, P., Ammar, M. H., Dhamdhere, A., Raj, H., Riley, G. F., and Zegura, E. W. Bootstrapping in Gnutella: A Measurement Study. In *PAM* (2004), vol. 3015 of *Lecture Notes in Computer Science*, Springer, pp. 22–32.

[5] Liu, Liu, Xiao, Ni, and Zhang. Location-aware Topology Matching in P2P Systems. In *INFOCOM: The Conference on Computer Communications, joint conference of the IEEE Computer and Communications Societies* (2004).

[6] Liu, Xiao, Liu, Ni, and Zhang. Location Awareness in Unstructured Peer-to-Peer Systems. *IEEETPDS: IEEE Transactions on Parallel and Distributed Systems 16* (2005).

[7] Lua, K., Crowcroft, J., Pias, M., Sharma, R., and Lim, S. A Survey and Comparison of Peer-to-Peer Overlay Network Schemes. *Communications Surveys & Tutorials, IEEE* (2005), 72–93.

[8] Lv, Q., Cao, P., Cohen, E., Li, K., and Shenker, S. Search and Replication in Unstructured Peer-to-Peer Networks. In *Proceedings of the 2002 International Conference on Supercomputing (16th ICS'02)* (June 2002), ACM, pp. 84–95.

[9] Papadakis, C., Fragopoulou, P., Athanasopoulos, E., Dikaiakos, M. D., Labrinidis, A., and Markatos, E. A Feedback-based Approach to Reduce Duplicate Messages in Unstructured Peer-to-Peer Networks. In *Integrated Research in GRID Computing*. February 2007.

[10] Saroiu, S., Gummadi, P. K., and Gribble, S. D. A Measurement Study of Peer-to-Peer File Sharing Systems. Tech. rep., July 23 2002.

[11] Stutzbach, D., and Rejaie, R. Capturing Accurate Snapshots of the Gnutella Network. IEEE, pp. 2825–2830.

[12] Stutzbach, D., Rejaie, R., and Sen, S. Characterizing Unstructured Overlay Topologies in Modern P2P File-sharing Systems. In *Internet Measurment Conference* (2005), USENIX Association, pp. 49–62.

[13] Zhenzhou, Z., Panos, K., and Spiridon, B. DCMP: A Distributed Cycle Minimization Protocol for Peer-to-Peer Networks. In *Parallel and Distributed Systems, IEEE Transactions* (2008), vol. 19, IEEE, pp. 363–367.