

Topology Awareness in Unstructured P2P Networks using Accessible Bandwidth

B. Lalitha
Department of CSE
JNTUCEA, Anantapur, India

Ch D V Subba Rao, Ph.D
Department of CSE
S V University, Tirupathi, India

ABSTRACT

The ever increasing growth of peer-to-peer networks calls for effective construction of overlays keeping in view the properties and capabilities of the underlay network. The capabilities of the peers participating in the network can be used effectively to bias the neighbor selection and improve the network performance by assigning more responsibility to nodes with high capabilities. This system considers the available bandwidth of peers for overlay formation in the Gnutella network. The designed system is tested with proper analysis and simulations to verify the functionality of the system. The simulated results show that the enhancements to Gnutella network improve performance of the network and results in better utilization of the system capabilities.

General Terms

Distributed Networks, Network Traffic, Simulation, Overlays.

Keywords

Peer-to-Peer, Gnutella, Topology Awareness, Bandwidth.

1. INTRODUCTION

Peer-to-Peer (P2P) networks are widely accepted and are growing at a rapid pace due the ability of the P2P networks to sustain the major distributed system networking problems such as scalability, heterogeneity and dynamic nature of the nodes present in the network. These P2P systems are utilized not only for distributed high performance scientific computing but also for general purpose applications as they are becoming easiest and cheapest way to share resources and increase the computing capacity. It is evident from the studies [1] that P2P is popular as 70% of the total Internet traffic is resulted P2P systems.

Although P2P networks provide many desirable properties for the users like anonymity, low maintenance costs and fault tolerance, they have some inherent problems in the current P2P networks:

- A. **Topology Mismatch Problem:** The inherent mismatch which exists between the P2P application layer and the underlying Internet due to the formation of overlays without having the underlay topological awareness results in redundant network traffic and low performance of the system.

Consider the P2P network depicted in fig 1 having 5 peers in the overlay network and 3 non participating

nodes along with overlay nodes in the underlay network.

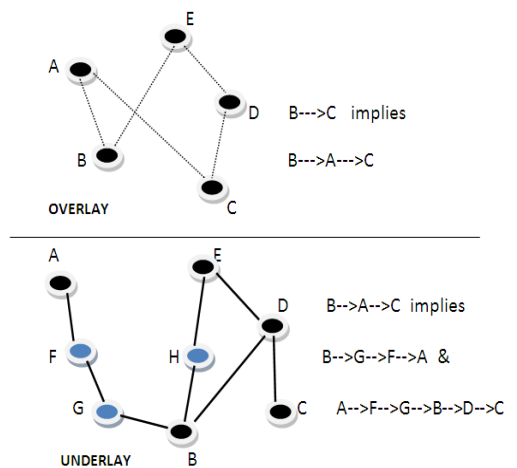


Fig 1: Topology Mismatch

In the overlay network each node has two nodes as neighbors. The random selection of peers as neighbors in Gnutella without considering the underlay network properties lead to an overlay with low performance guarantee and redundant messages.

In the network if a data item is to be sent from B to C then the overlay routing involves the path BAC. Since the actual routing occurs at the underlay level the actual data transmission path is BGFAFGBDC. As the resultant path shows that the routed path is not optimal for the data transmission. This is the topology mismatch problem where the overlay topology formation does not consider the underlay network properties like bandwidth, latency, location etc.

- B. **Inefficient Bandwidth Utilization:** The P2P applications like Gnutella uses the mechanism of flooding or pure broadcast for query search which consumes a lot of bandwidth due to the number of query messages generated in the network. Generally bandwidth is a property between two machines but practically the bandwidth between a peer and the rest of internet is determined by the nodes direct

link to the internet. As a result the bandwidth is a considered as a property of the peer.

Studies [5] show that the average upstream bandwidth of nodes in a Gnutella network is around 1Mbps, while 22% of peers have bandwidth below 0.1Mbps, 8% of peers have bandwidth above 10Mbps, and the highest observed bandwidth capacities are around 100Mbps. This inbuilt heterogeneity in the Gnutella network affects the performance of the system if all the nodes are treated equally.

The system presented here tries to solve the above two problems with more focus on the bandwidth utilization as the bandwidth improvements in the system will have outstanding affects on the speed and performance of the system and also recent trends reveal that companies are more interested to improve the performance of the existing systems by monitoring and efficient utilization of bandwidth rather than increasing the amount of bandwidth due to high costs and resource consumption by the distributed devices.

This paper presents the bandwidth aided topological awareness to improve the performance of the unstructured P2P systems like Gnutella by building a highly scalable system which avoids overloading any of the nodes by accounting for the capacity of nodes. The rest of the paper is structured as follows. Section 2 specifies the related areas which led to proposal of this system. Section 3 introduces the overlay topology formation. The Section 4 illustrates the query search mechanism implemented. Section 5 presents the simulation and numerical results. Finally, we summarize in Section 6.

2. RELATED WORK

The older versions of Gnutella i.e. Gnutella 0.4 [2] uses the flooding-based search to find files in the P2P network. To avoid the drawbacks of flooding mechanism the proposed system uses the biased random walks in which the search query is biased towards high capacity nodes which are best to answer the queries. The biased random walks generate less number of messages compared to flooding which reduces the overhead on the network.

The super node approach was made popular by the introduction of KaZaA [6] which helped to built P2P systems that were scalable. Due to its success in the implementation this approach is introduced in the Gnutella systems.

The present stable version of Gnutella i.e. Gnutella 0.6 [3] classifies the peers into ultra-peers and leaves based on the bandwidth and capacity of the peers. Unlike Gnutella 0.6, in the proposed system there is no categorization of peers, but the peers are assigned a capacity value which is used in connection establishment and query forwarding. This system allows for even load balancing by adapting the duties of the peers based on the available bandwidth of the peers.

Studies [4] show that nodes in the Gnutella network show significant heterogeneity in terms of their capabilities such as bandwidth, online times, hard disk space and processing

power. Gnutella 0.6 tries to solve this problem using the multi level topology formation, but the problem is in the ultra-peer election process involves the decision of which peer to become the ultra-peer and configure the total number of ultra-peers for scalability which involves complex issues to deal with. The proposed system in this paper considers the inherent heterogeneity and achieves better performance and scalability.

Notable solutions to exploit the heterogeneity in the Gnutella network are mentioned in [7] [8] [9]. The methods described in [7] [9] provides a way to assign more responsibility to some high capacity nodes to provide better scalability and performance in the Gnutella systems. The solution in [8] replaces the Gnutella search with random walk to reduce the overhead in the network.

Gnutella Systems are not scalable when a large number of queries are introduced into the network. In such cases the nodes in the network gets overloaded and system performance is affected. This problem gets even worse when the system size increases. The proposed system handles the higher rate of queries with increasing system size by avoiding the overload at any of the nodes in the network.

3. OVERLAY TOPOLOGY FORMATION

In general the capacity of a node is characterized by the number of queries that the node can handle per second. In general for low-bandwidth nodes the query processing capacity is determined by the nodes available bandwidth. On the other hand for the nodes with higher speed connection, along with bandwidth other properties such as speed of the CPU, disk latency etc. will affect the node's capacity. The bandwidth-aided system implementation ignores the affects of CPU speed, disk latencies and the capacity is assumed as a function of available bandwidth. The overlay topology formation of the system is described as follows:

When a node enters the network it uses the bootstrapping mechanism similar to the one found in Gnutella to find the other nodes of the network. Each peer maintains a cache containing the list of all other peers present in the network. The cache entries will contain the IP address, port number and the capacity values of the other peers. These caches are exchanged periodically with the neighbors and updated accordingly. The old and stale entries are marked as dead.

The main aim of the overlay topology formation is to ensure that high capacity nodes have higher degree and the lower capacity nodes are at short reach of lower capacity nodes. To achieve this purpose each node computes a *stability level*. This quantity is a function node's current set of neighbors, their capacities, degree and preconfigured maximum and minimum number of neighbors. This measure takes values between 0 and 1 that signifies how stable the node is with the current set of neighbors. As long as the node is not stable, it will continuously configure its neighbors and improves its level of stability. The stability value can be given as follows:

$$S(X) = \frac{\sum_{i \in Nbr_x} \frac{C_i}{Nbr_i}}{C_x}$$

The measure $S(X)$ returns a value between 0 and 1. If the value is 0, the node X is not stable and continues to search for neighbors with better capacity values. If the value is 1, the node X is stable and only tries to calculate the stability value periodically to keep the node stable.

When a node starts up it has minimum preconfigured set of neighbors and it is assigned the maximum number of neighbors it can have based on its bandwidth. Now the node starts gathering more neighbors and improves its level of stability. The node will continuously reconfigure its neighbors until the node will have optimal set of neighbors. This process is described in algorithm 1.

To add a new neighbor, a node randomly selects a small set of entries from its own cache that are not already neighbors or dead entries. From these entries the node selects those nodes with greater capacity than itself and sends the connection request message. If no such eligible entry exists the node randomly selects one entry from the cache and sends the connection message.

Let $C_i \leftarrow$ Capacity of Node i
 $Nbr_A \leftarrow$ Number of Neighbors of A
 $Max_Nbr_A \leftarrow$ Maximum neighbors Node A
can have
 $Min_Nbr_A \leftarrow$ Minimum Neighbors Node A
should have
 $S \leftarrow$ Stability Value
If $Nbr_A < Min_Nbr_A$ then
Return 0
Value $\leftarrow 0$
 $\forall X \in Nbr_X$ do
Value \leftarrow Value + $\frac{C_x}{Nbr_x}$
 $S \leftarrow \frac{Value}{C_x}$
If $S > 1$ or $Nbr_A > Max_Nbr_A$ then
 $S \leftarrow 1$
Return S

Algorithm 1: Level of Stability

During the connection establishment phase each node makes decision of whether to accept the other node as neighbor or not based on its capacity with the existing neighbors and with the new one. To accept the new node, an existing neighbor will be dropped based on the capacity value. The neighbor acceptance is based on the maximum number neighbors that the node can accept. While dropping the neighbors, the degree of the dropping node is also taken into consideration in order to avoid dropping connections of already poorly connected nodes in favor of highly connected ones. The process of neighbor adaptation will continue until a stable state is reached at each node. The neighbor elimination process is described in algorithm 2.

The idea behind use of capacity is that a node with capacity C will forward approximately C messages per second at its full load and it needs adequate outgoing capacity in all its neighbors to maintain that load. In addition to the factors discussed, a number of other properties may be used to compute the stability levels such as the load on the neighbors, network locality etc. For simplicity this system implementation uses the capacity of the nodes and degree to compute the stability level.

Once the node is fully stable, it will still continue to perform the neighbor reconfiguration process checking its stability levels every T seconds by keeping in mind the mobility of nodes.

Let $C_i \leftarrow$ Capacity of Node i
 $Nbr_A \leftarrow$ Number of Neighbors of A
 $Max_Nbr_A \leftarrow$ Maximum neighbors Node A
can have
 $Min_Nbr_A \leftarrow$ Minimum Neighbors Node A
should have
If $Nbr_A + 1 \leq Max_Nbr_A$ then
Accept B ; return
Else
Candidates $\leftarrow x \forall X \in Nbr_X$ such that
$C_x \leq C_B$
If NotExists then
Reject B
Return.
 $M \leftarrow$ Neighbor with highest degree
from candidates
If $(C_B > \max(C_X \forall X \in Nbr_X))$ Or
 $(Nbr_A > Nbr_B)$ then
Drop M ; Accept B
Else Reject B .

Algorithm 2: Neighbor Elimination

The addition of new neighbors is depicted in fig 2. The existing network contains the nodes A, C, D, E and F . Now node B sends a request to A to join as neighbor, then A will add B as its neighbor if the room exists. If there is no room then it checks all the capacities and degrees of its neighbors then decides the request of new node. Among the neighbors of A if there exists a node with lower capacity than B and with higher degree then A will add B as its neighbor and drop the lower capacity node. In the figure if the capacity of Node D is lower than B , then B will be accepted as neighbor since D has good degree to continue its operations.

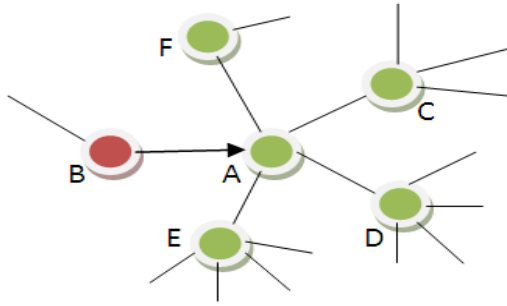


Fig 2: Adding New Neighbor

In addition to the cache the nodes maintain an index for the data stored in each of its neighbor nodes. The indices are exchanged when neighbors establish connections to each other. These indices are useful for query response for not only its data but also its neighboring data. When the neighbor leaves the system or the topology is updated the index information for that neighbor node is removed.

4. SEARCH PROTOCOL

Unlike Gnutella, this system uses the biased random walk for searching the data items in the P2P network. Biased random walk is the method of forwarding the incoming query to highest capacity neighbor instead of forwarding to a random neighbor of all the neighbors.

In order to limit the random walk at some point of time, the TTL values are used their by limiting the number of hops a query can progress. To avoid redundant paths the nodes store the QID of the query and the neighbors to which it has already forwarded the query, which reduces the chance that a query is forwarded twice in the same path.

The query is resent and forwarded until the minimum numbers of responses for the query are received from the nodes. Since a node can respond to not only its own data but also its neighbors the response is made with the identification of the node having the data.

The search protocol is depicted in the fig 3. It shows how the query is forwarded using the biased random walk where the bias is towards the high capacity nodes, which are represented using larger circles.

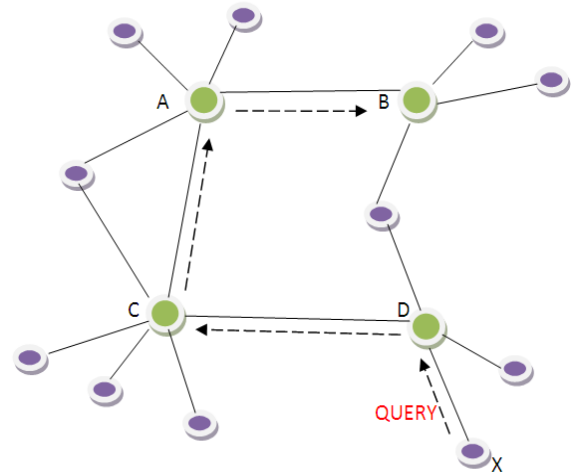


Fig 3: Query Search

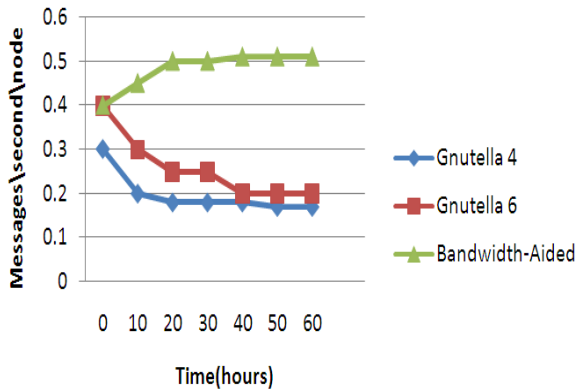
In the figure Node *X* will generate the query, which is forwarded to the neighbor with highest capacity i.e. *D*. Node *D* will forwarded to its highest capacity neighbor i.e. *C*. In this manner the query is forwarded to *A* then *B*. Since the query is reached to many high capacity node there is high probability that many responses will be generated by these nodes. And also since the nodes can also replay to the data present in its neighbors, the high capacity nodes can provide many responses compared to other nodes.

5. SIMULATION

This section provides the details about the simulation setup and the results that support the improvements of performance of the system. First each node in the system is assigned a capacity value which represents the number of messages it can accept per unit time. The capacities are assigned based on Gnutella distributions derived from [5]. In addition the nodes are given a query generation rate value which represents the number of queries that the node generates per unit time.

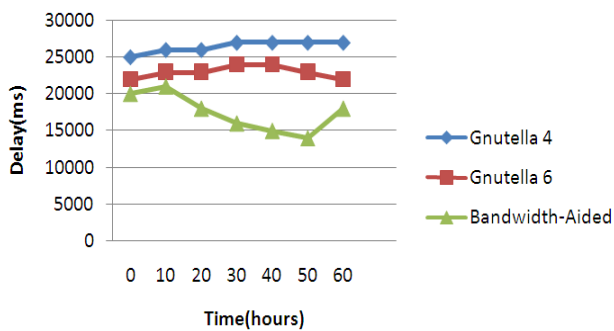
In order to evaluate the effectiveness of the proposed system, we compare the simulate results with that of Gnutella 0.4 and Gnutella 0.6. Each node in the Gnutella 0.4 system is modeled with 4 neighbor nodes. In the Gnutella 0.6 each ordinary peer has 3 ultra-peer connections and each ultra-peer has 10 ultra-peers as neighbors. The node in the proposed system is configured with minimum of 3 neighbors and the maximum is based on the capacity of nodes. On an average the nodes in the bandwidth aided systems have 20 neighbors.

The metrics taken into account for evaluation are maintenance overhead, query delay and query success rate. These metrics are considered for all the three networks and the results are represented in the following graphs.



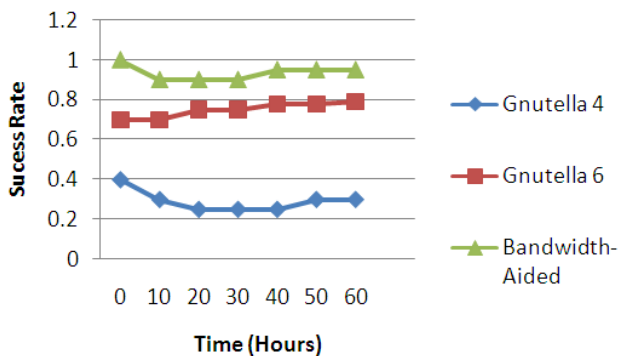
Plot 1: Maintenance Overhead

The plot 1 depicts the maintenance overhead in the network. It shows that the Bandwidth aided system has higher maintenance overhead at the node as the nodes have to maintain the cache and indices for all the neighbors in the network. But the drawback of maintenance overhead is compensated in the form of faster query results, as the nodes respond to the queries faster due to the presence of indices of neighbors, as a result the delay for queries is less which is depicted in plot 2.



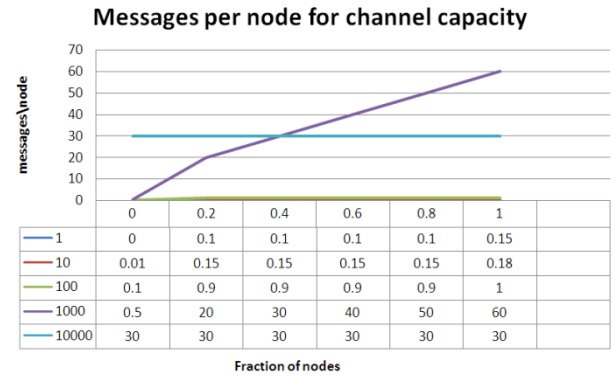
Plot 2: Query Delay

The bandwidth aided system's query success rate is also very high compared to the Gnutella networks. This is represented in plot 3.



Plot 3: Query Success Rate

To better understand the distribution of load among the nodes, we consider the load at the nodes for each of capacity values. This is depicted in plot 4.



Plot 4: Messages per Node for Channel Capacity

The plot shows that the capacity 100 nodes introduce higher overhead compare to capacity 1 and 10 nodes but less than capacity 1000 nodes.

Since the nodes of bandwidth-aided network have higher in-degree, these will be often visited by the random walk which increases the success rate and in turn reduces the average message overhead on the network. A bandwidth-aided system overhead without queries is nearly twice that of Gnutella 0.6, but these are comparable with queries due to shorter random walks.

6. CONCLUSION

The Bandwidth-Aided topology awareness system is a highly scalable unstructured P2P system which improves the performance of the Gnutella network by considering the peer bandwidth for the topology formation and distributing the load on the high capacity nodes. The system makes the search more scalable and reduces the query response time which resulted in drastic improvement in the query success rate with small overhead. The future work for this system would be considering other underlay properties of the peer while constructing the overlay.

7. REFERENCES

- [1] H. Schulze and K. Mochalski. Data about P2P, VoIP, Skype, file hosters like RapidShare and streaming services like YouTube. Ipoque, Internet Study 2007, November 2007.
- [2] The Annotated Gnutella Protocol Specification 0.4, <http://rfc-gnutella.sourceforge.net/developer/stable/index.html>.
- [3] Tor Klingberg and Raphael Manfredi. Gnutella 0.6, 2002. http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html.
- [4] Sariou, S., Gummadi, P. K., and Gribble, S. D. "A Measurement Study of Peer-to-Peer File Sharing Systems. In Proceedings of Multimedia Computing and Networking 2002" (MMCN'02) (San Jose, CA, Jan. 2002).

- [5] S. Saroiu, P. K. Gummadi, and S. D. Gribble. “Measuring and analyzing the characteristics of napster and gnutella hosts”. *Multimedia Systems*, 9(1):170-184, July 2003.
- [6] KaZaA, 2002. <http://www.kazaa.com/>.
- [7] LV, Q., Ratnaswamy, S., AND Shenkar, S. “Can Heterogeneity Make Gnutella Scalable”. Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02). (Cambridge, MA, Mar. 2002).
- [8] LV, Q., CAO, P., COHEN, E., LI, K., AND SHENKER, S. “Search and Replication in Unstructured Peer-to-Peer Networks “. Proceedings of 16th ACM International Conference on Supercomputing 2002.
- [9] Srivatsa, M., Gedik, B., Liu, L.: “Large Scaling Unstructured Peer-to-Peer Networks with Heterogeneity-Aware Topology and Routing”. IEEE Transactions on Parallel and Distributed Systems 2006.