

Search Performance Analysis in Peer-to-Peer Networks

Tsungnan Lin, Hsinping Wang
Dept. of Electrical Eng.
National Taiwan University, Taipei, Taiwan
tsungnan@ntu.edu.tw

Abstract

Recently Peer-to-Peer networks (P2P) have gained great attention and popularity. One key challenging aspect in P2P resource sharing environments is efficient searching algorithm. This is especially important for Gnutella-like decentralized and unstructured networks since they have power-law degree distributions. A robust search algorithm should respond to the query message promptly without generating redundant query messages. We present unified quantitative search performance measurements: Query Efficiency, Search Responsiveness, and Search Efficiency to objectively capture dynamic behaviors of various search algorithms from different perspectives. To gain insight of these search algorithms, we quantitatively characterize, through simulations, their search performance on different network topologies with different query/replication distributions.

1. Introduction

In this paper, we propose quantitative measure criteria of searching performance: Query Efficiency, Search Responsiveness, and Search Efficiency. Search Efficiency, the multiplication of Query Efficiency and Search Responsiveness, tries to give an objective performance measure from both users' and networks' perspectives by taking several performance metrics into consideration. These factors include number of results found, success probability, search speed, and total number of messages.

Our results show that current Gnutella searching algorithm [1] gives ideal search efficiency when the search range is limited to local. It is not surprising since it sends query messages to every possible node and can return every possible result as quickly as possible. The search efficiency decays exponentially as the search time increases since the number of query messages increases linearly with the size of visited peers. Our results indicate the flooding search algorithm faces the scalability problem [2, 3] when the query time increases.

It has been suggested random walker search algorithm [10, 12] can improve the scalability problem. We find that the search efficiency of random walker algorithm almost remains the same regardless of the search time because

the number of the query messages (walkers) remains constant regardless of the network topology. However, random walker search algorithm suffers from poor search efficiency in the short term although it does have higher search efficiency compared to that of flooding search algorithm in the long term. Besides, it is difficult to determine the optimal number of walkers in a dynamic environment in advance.

2. P2P environment setup

The measurement in [6] has suggested that the topology of Gnutella network has the property of two-stage power-law distribution. Therefore, simulations are performed in a network consisting of 10,000 nodes and the link distribution of the network follows the measure characteristics reported in [6]. The maximum link degree is 199 with mean of 6.05 and standard deviation of 13.09. We assume there are 100 distinct objects with 100 replications each; totally there are 10,000 objects in the network. We set that 25% of nodes sharing nothing (this illustrates the facts that Gnutella has an inherently large percentage of free-riders), 35% of nodes sharing only one object, and only 1% of nodes sharing more than six objects.

3. Search efficiency

A good search algorithm must be scalable, efficient, and responsive. In this section, we propose a unified search analysis criterion to evaluate the quality of search algorithms in terms of scalability, efficiency, and responsiveness.

An efficient search algorithm should not generate a huge number of redundant messages in an uncontrolled fashion and overwhelmingly waste the network bandwidth unnecessarily. In addition, an efficient algorithm means that the query messages generated during the search process should have a high hit rate (finding the target objects). Therefore, we define "Query Efficiency (QE)" as the ratio of Query Hits to Messages Per Node:
$$\text{Query Efficiency} = \frac{\text{Query Hits}}{\text{QueryMsg}/\text{NetworkSize}} = \text{QueryHits} / \text{MsgPerNoe}.$$

Another important factor in the search performance is Search Responsiveness, evaluating responsiveness and reliability. Responsiveness is the ability of a search algorithm to respond quickly to meet the needs of a user. In other words, a responsive algorithm is the one with a fast lookup mechanism. Additionally, reliability is essential to a healthy responsive algorithm. Reliability means the ability for a search algorithm to meet the commitments made to users. When a query is issued, it should be the highest priority of a search algorithm that the commitment (successfully finding the target) should always be met. Therefore, “*Search Responsiveness (SR)*” measuring the responsiveness and reliability of a search algorithm can be defined as: $Search\ Responsiveness = SuccessRate / HopsNumber$.

To capture the characteristics of efficiency and responsiveness of search algorithms, the unified criteria “*Search Efficiency*” can be defined as $Search\ Efficiency = Query\ Efficiency \times Search\ Responsiveness = (QueryHits \times SuccessRate) / (MsgPerNode \times HopsNumber)$.

3.1 Search efficiency analysis of experiments

Search Efficiency, Query Efficiency, and Search Responsiveness of various search algorithms are shown in Figure 1. From the Figure 1(b) of Query Efficiency, we can see Query Efficiency of flooding algorithm will decay dramatically with respect to the search time since the number of query messages grows exponentially and the number of query messages increases at much higher rate than that of the number of query hits. Additionally, QE of expanding ring algorithm falls below that of flooding algorithm. Since expanding ring algorithm stops searching whenever a target is found, the number of query hits is low and the volume of redundant messages in the local area is high. However, Query Efficiency of random walk algorithms remains almost constant when the search time increases. Although the total number of query messages grows linearly as the search time increases, the number of query hits also increases. Therefore, QE of random walk algorithm keeps constant regardless the search time. From the figure, it is noted that QE of random walk algorithm is four times better than that of flooding algorithm.

From the Figure 1(c) of Search Responsiveness, we can easily see flooding algorithm generates the fastest response since it aggressively sends the query messages. It is not surprising to see SR of random walk algorithm gives the slowest result. From the Figure, it is interesting to know that the speed of flooding algorithm is about 2.4 times faster than that of random walk.

The overall performance (Search Efficiency), as displayed in Figure 1(a), can be obtained from the multiplication of QE and SR. Although random walk algorithm has the best QE performance, the performance

of SE is not satisfactory because of the low speed. SE of flooding algorithm is high in the short term, it, however, will decay dramatically due to the huge amount of redundant messages when the search time increases. According to the discussion and observation above, we conclude that flooding is a responsive algorithm and random walk is an efficient one.

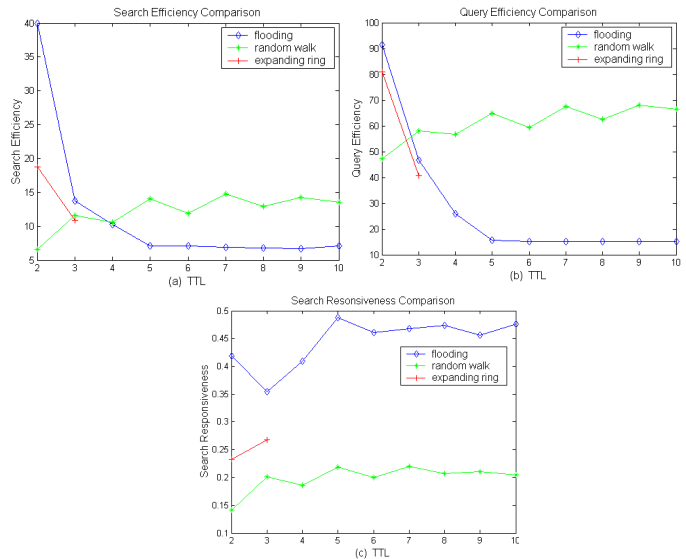


Figure 1. Search Efficiency, Query Efficiency, and Search Responsiveness comparison of the three algorithms simulated in Gnutella network

4. Conclusion

Our results show current search algorithms either overwhelm the whole network bandwidth hoping to meet users’ satisfactory requirement, or sacrifice the responsive performance in order to produce scalable solutions. Flooding algorithm generates the best performance in terms of Search Responsiveness but its Query Efficiency is low due to huge number of redundant messages. Random walk algorithm enjoys high Query Efficiency but suffers from low Search Responsiveness.

5. References

- [1] The gnutella protocol specification v0.4. http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf.
- [2] K.Sripanidkulchai, The popularity of Gnutella Queries and its Implications on Scalability, white paper, Carnegie Mellon Univ. Pittsburgh, Feb. 2001.
- [3] Clip2. Gnutella: To the bandwidth barrier and beyond. <http://www.clip2.com/gnutella.html>, 2000.
- [4] Q. Lv, P. Cao, E. Cohen, E.Felten, X. Li and S. Shenker. Search and replication in unstructured peer-to-peer networks. Proc. 2002 ACM SIGMETRICS, 2002.
- [5] Lada Adamic, R. Lukose, and B. Huberman. Local Search in Unstructured Networks. Handbook of Graphs and Networks:

From the Genome to the Internet, S. Bornholdt and H.G.
Schuster (eds.), Wiley-VCH, Berlin, 2000.
[6] L. Adamic, R. Lukose, A. Puniyani, and B. Huberman.
Search in Power-Law Networks. Phys. Rev. E, Vol. 64, pages
46135-46143, 2001.