Dynamic Search in Peer-to-Peer Networks

Hsinping Wang ,Tsungnan Lin, Chia Hung Chen, and Yennan Shen
Graduate Institute of Communication Engineering
National Taiwan University, Taipei, Taiwan
+886-2-23635251-536
{r91942041, tsungnan}@ntu.edu.tw

ABSTRACT
This work* specifically addresses the search issues in unstructured peer-to-peer (P2P) systems that involve the design of an efficient search algorithm, the proposed dynamic search, and the modeling of P2P systems reflecting real measured P2P networks. Through simulations, we will show dynamic search outperforms other existing ones in terms of performance aspects.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Search Process.*

General Terms: Algorithms, Performance, Design

Keywords: P2P, Search algorithm, Gnutella, Modeling

1. INTRODUCTION
One key challenging aspect in P2P systems is the design of efficient search algorithms. It is especially important for Gnutella-like unstructured P2P networks since they have highly-connected power-law topologies that easily lead to in-scalable operation. Thus, we propose a scalable search, which dynamically decides the number of running walkers with respect to peers' local topology and search time state. It is able to temporally control the extent of messages generated by the simulated annealing mechanism, thus a scalable one. Moreover, we quantitatively characterize, through simulations in dynamic P2P environments, the performance of various existing algorithms. The proposed dynamic search outperforms other algorithms in terms of *SE* [1] in both the local and global search spaces.

2. DYNAMIC SEARCH
The design goal of our Dynamic Search algorithm is to optimize the search performance. We hypothesize that the intrinsic nature of search optimization methods in an unstructured P2P network is similar to the problems of the combinatorial optimization methods. Therefore, we take an approach similar to the mechanism called simulated annealing, a popular optimization method, to construct our dynamic search. The term simulated annealing derives from the roughly analogous physical process of heating and then slowly cooling a substance to obtain a strong crystalline structure, which allows the system to move consistently towards lower energy states, yet still jump out of local minima due to the probabilistic acceptance of some upward moves during the first few iterations.

We specify our dynamic search by a forwarding probability model. For each peer applying dynamic search, the probability function to forward to its *i*th neighbor at search time *t* is

$$p_i(t) = \underbrace{p^0 \cdot u(t) - p^1 \cdot u(t-1) - \cdots\cdots - p^n \cdot u(t-n)}_{Annealed\_Flooding} + p^u \cdot u(t-n) \quad (1)$$

where n is the limit to aggressively explore the search space. The probability p^0 controls the extent the peer to query its neighbors and probabilities p^0, p^1,…, and p^n *anneal* the querying extent. To guarantee that the number of query messages will not grow exponentially, p^u can be set to be $1/l$, where l is the link degree of the relayed node. In this case, only one query message will be generated.

Dynamic search is hence a multi-stage search with a probability model annealed to search time. It attempts to probabilistically flood the local search space and temporally limits the exploration. While the search is out of the local space (beyond n hops), dynamic search will change to a limited search. The multi-stage probability model can satisfy the goals to search aggressively in the local, control the network impact in the global, and choose the number of optimal query messages in a dynamic fashion. Furthermore, by the dynamic search, peers can probabilistically decide the number of query messages (walkers) according to their linking status and the time state t within radius n, while random walk can only use fixed number of walkers.

3. REALISTIC P2P MODELING
In order for strong evaluation environment, we have built a simulator, with algorithms in question embedded, to model possible aspects in file-sharing P2P systems: network topology, peer cycle, and peer querying/sharing behavior. The significance of this simulator comes from these modeled aspects that strongly reflect the real measurement studies [4, 5, 6], and the dynamic modeling of peer cycle, thus producing virtually realistic results.

For topology modeling, our simulator constructs a P2P network of 100,000 nodes and the link distribution follows the two-stage power law, reported in [6]. The maximum link degree is 632 with mean of 11.73 and standard deviation of 17.09.

The peer cycle is modeled as joining, querying, idling, leaving, and joining again to form a cycle, which is similar to the one in [3]. The joining and leaving operations (include idling) are inferred and then modeled by the uptime and session duration distributions measured in [4, 5]. We use log-quadratic distributions suggested in [5] to re-build the two distributions so that 1) half of the peers have uptime percentage less than 10% and the best 20% of peers have 45% uptime or more, and 2) the median session duration time is 20 minutes. By the rebuilt distributions, we can generate a probability model to decide when a peer should join or leave the network and how long it should continually be online. The basic rule is that

* This work is supported in part by NSC grant NSC92-2213-E-002-087.

WWW 2004, May 17-22, 2004, New York, NY USA.
ACM 1-58113-912-8/04/0005.

nodes with higher link degrees are assigned to higher uptime percentages and longer session durations, and vice versa. With these conditions, we map a *two-hour* long dynamic join/leave pattern for peers. On average, there are 10 peers joining or leaving simultaneously. The average number of on-line nodes is 18,152 with maximum of 24,218 and minimum of 4,886.

We model peer querying as Poisson distribution with mean idle time λ= 50 minutes—one query per 50 min. per peer, which is an experiential value [3]. Thus, there are statistically 6 queries or searches processing concurrently in the entire network. Totally, in this 2-hour simulation, we generate 43,632 search queries.

For the object distribution, we assume there are 100 distinct objects with replication ratio R = 1%; totally there are 100,000 objects in the network. The distribution of the 100,000 objects over the network follows the characteristics reported in [4]. For the peer query distribution of these 100 distinct objects, we model it as zipf distribution with parameter $a = 0.82$ [2, 6]. Finally, our simulator's central clock is triggered per second, which measures a hop for messaging passing.

4. PERFORMACE EVALUATION

Before presenting our simulation, we first introduce a unified search evaluation metric, *Search Efficiency* [1], which measures scalability, reliability, and responsiveness in a single shot:

$$Search\ Efficiency\ (SE) = \frac{QueryHits \times SuccessRate}{MsgPerNode \times HopsNum}.$$

We compare four search algorithms (flooding, random walk, modified-BFS [2], and dynamic search) by *SE*. The number of walkers for random walk is chosen as 10. The fraction parameter of modified-BFS is 0.2. Parameters in (1) are set as $n = 2$, $p^0 = 1.0$, $p^1 = 0.7$, $p^2 = 0.3$, and $p^u = 1/l$, where l is the link degree of relayed peer. With these specifications, the dynamic search will operate as flooding at hop count h = 1, as modified-BFS with probability of 0.3 at h = 2, and as random walk for h≧3.

Simulation environments are entirely described in Section 3. In brief, we perform a 2-hour simulation, issuing 43,632 searches for each algorithm. The simulation results for *SE* with hop depths h = 1 to 7 are plotted in Figure 1.

We can observe in Figure 1 that *SE* of flooding is high compared with random walk in the local space (h = 2) but decrease dramatically in the global region. Modified-BFS sustains the high efficiency from h = 3 to 4 but performs poorly in the long-term space. The performance degradation of these two algorithms is due to the huge number of redundant messages, as discussed in [2, 6]. Random walk, on the other hand, needs some "warm up" time to explore the search space and the performance increase is delayed to the long-term search space due to the limited number of random walkers. In the long term, its SE is consistently high compared with flooding and modified-BFS. As for the dynamic search, it inherits the high performance of flooding in the local and the consistent performance of random walk in the global, thus performing outstanding performance within all hop depths. Moreover, in Section 2, we claim that our dynamic search decides the number of walkers each node forwards to the next-hop peers in a "simulated annealing" style. We therefore show the analysis of walker numbers on a hop-depth basis in Figure 2, in which we make a metaphor relating the number of walkers with the temperature of annealing systems.

For flooding, the apparent large number of walkers "overheats" the search system (overwhelms it by messages) since it lacks the cooling process essential for the optimization. For random walk, the initial number of walkers is 10, but it imposes a hard limit on the walker number as 1 when h□2, which corresponds to the lack of chance to jump out of local minima, thus resulting in a too "cool" state of annealing and this is why it "warms up" slowly. But for our dynamic search, the number of walkers is 16.2 initially, annealed to 7.5, and finally kept a constant of 1. In this way, dynamic search aggressively explores the local, giving chance to jump out the local minima, to return responsive query hits and increase the success probability. In the global search space, it moderately controls messages generation, applying the cooling process, thus leading to consistent search performance.

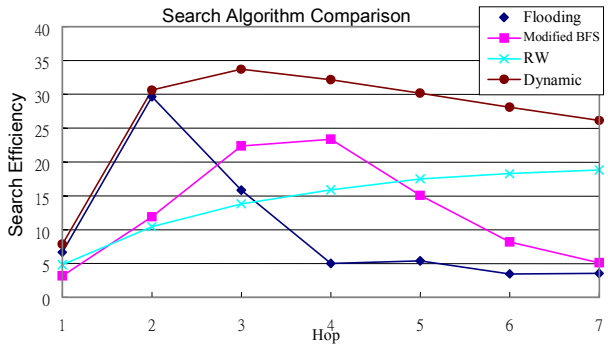

Figure 1. Search Efficiency comparison for search algorithms

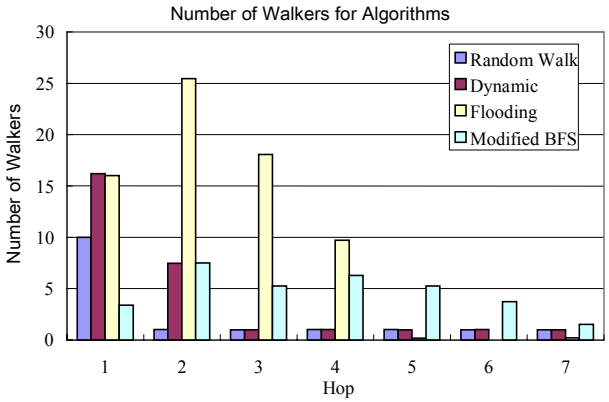

Figure 2. The average number of walkers each peer generates at various hop-depths for algorithms

5. CONCLUSION

In this paper, we have demonstrated two major contributions: 1) the design of dynamic search, which adopts the ideas of simulated annealing to achieve good performance, and 2) a dynamic P2P system modeling, which strongly reflects the real P2P measurements, thus giving convincing results to support the fineness of the dynamic search.

6. REFERENCES

[1] T. Lin, H. Wang, and J. Wang. Search Performance Analysis and Robust Search Algorithm in Unstructured Peer-to-Peer Networks. GP2PC, April 2004.

[2] D. Tsoumakos and N. Roussopoulos. A Comparison of Peer-to-Peer Search Methods. WebDB, June 2003.

[3] Z. Ge, D. R. Figueiredo, S. Jaiswal, J. Kurose, and D. Towsley. Modeling Peer-Peer File Sharing Systems. INFOCOM, 2003.

[4] S. Saroiu, P. K. Gummadi, and S. D. Gribble. A measurement study of peer-to-peer file sharing systems. MMCN, January 2002.

[5] J. Chu, K. Labonte, and B. Levine. Availability and Locality Measurements of Peer-to-Peer File Systems. SPIE, 2002.

[6] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. Search and replication in unstructured peer-to-peer networks. ICS, 2002.