Unstructured Peer-to-Peer Networks: Topological Properties and Search Performance

George H.L. Fletcher*, Hardik A. Sheth**, and Katy Börner***

*Computer Science Department
** School of Informatics
*** School of Library and Information Science,
Indiana University, Bloomington, USA
{gefletch, hsheth, katy}@indiana.edu

Abstract. Performing efficient decentralized search is a fundamental problem in Peer-to-Peer (P2P) systems. There has been a significant amount of research recently on developing robust self-organizing P2P topologies that support efficient search. In this paper we discuss four structured and unstructured P2P models (CAN, Chord, PRU, and Hypergrid) and three characteristic search algorithms (BFS, k-Random Walk, and GAPS) for unstructured networks. We report on the results of simulations of these networks and provide measurements of search performance, focusing on search in unstructured networks. We find that the proposed models produce small-world networks, and yet none exhibit power-law degree distributions. Our simulations also suggest that random graphs support decentralized search more effectively than the proposed unstructured P2P models. We also find that on these topologies, the basic breadth-first search algorithm and its simple variants have the lowest search cost.

1 Introduction

Peer-to-Peer (P2P) networks have sparked a great deal of interdisciplinary excitement and research in recent years [17]. This work heralds a fruitful perspective on P2P systems vis-á-vis open multi-agent-systems (MAS)[1] [14]. A central issue for both P2P networks and MAS is the problem of decentralized search; an effective search facility that uses only local information is essential for their scalability and, ultimately, their success. Initial work on this issue suggests that there is a strong relationship between network topology and search algorithms; several deployed P2P networks [3,10,11,18] and MAS [2] have been shown to exhibit

[1] In an *open* MAS, agents do not have complete global knowledge of system membership.

G. Moro, S. Bergamaschi, and K. Aberer (Eds.): AP2PC 2004, LNAI 3601, pp. 14–27, 2005.

power-law degree distributions[2] and small-world properties.[3] In a small-world network, there is a short path between any two nodes. This knowledge, however does not give much leverage during search for paths in small-world systems because there are no local clues for making good choices. What is the best we can do for decentralized search in a small-world? There has been little comparative analysis of unstructured P2P models and search algorithms. Such validation and comparison of models and algorithms is the first step in answering this question.

The approach we have taken to explore this issue is to model the network topologies of two typical unstructured P2P models developed in the P2P community (PRU [19] and Hypergrid [21]) in simple graph-theoretic terms and build simulations of these networks to measure topological properties and search performance. As a comparison, we performed the same analyses on a random graph [3,18] and two structured P2P models (CAN [20] and Chord [24]). We show through these simulations that unstructured P2P networks have exactly the properties and problems of small-world topologies; the networks have low diameter but no means of directing search efficiently. Interestingly, these simulations also show that none of the models considered generate power-law degree distributions. This turns out to be desirable in an engineered system; although power-law networks support efficient decentralized search [1], they are fragile in the face of attack [3] and can unfairly distribute network traffic during search [21]. The reason for these weaknesses lies in the degree distribution; such networks have a few nodes of very high degree that serve effectively as local "hubs."

1.1 P2P Concepts and Related Work

There are two broad categories of P2P systems: hybrid and pure [17]. Hybrid systems are characterized by some form of centralized control such as a name look-up service [17] or a middle agent [8]. Pure systems strive for self-organization and total decentralization of computation; these systems are the focus of the work presented in this paper.

Pure P2P networks can be classified by the manner in which decentralization is realized. In *structured* systems [20,24], placement of system resources at nodes is strictly controlled and network evolution, consequently, incurs extra overhead. Ideally, one would strive to minimize system constraints and costly datastructures when designing a P2P model. *Unstructured* systems are characterized by a complete lack of constraints on resource distribution and minimal network growth policies. These systems focus on growing a network with the desirable low diameter of small world systems using only limited local information.

Early work on search methods for small world networks was done by Walsh [27] and Kleinberg [13] and on decentralized search in scale-free networks by

[2] The degree distribution of nodes in a graph follows a power-law if the probability $P(k)$ that a randomly chosen node has k edges is $P(k) \propto k^{-\tau}$, for τ a constant skew factor [3,18].

[3] A small-world network is characterized by low diameter and high clustering coefficient, relative to a random graph of equivalent size [28]. We will define these properties in full below.

Adamic et al. [1]. An early study of unstructured P2P network search performance was done by Lv et al. [15], comparing search performance on generic power-law, random, and Gnutella networks.[4] More recently, several groups have continued to study search performance with a focus on comparing power-law and random topologies with deployed P2P systems such as Gnutella [5,25,29]. Initial studies on search in open MAS have also focused on generic topologies [9,23]. Several projects have investigated the topological characteristics of the Internet [10] and implementations of P2P filesharing networks [11]. What has been missing in all of this work is a general comparative study of proposed unstructured P2P models, their topologies, and performance of search algorithms. This paper is an initial step in filling this gap in our understanding of decentralized search in unstructured P2P networks and open MAS.

2 P2P Models

In this section, we briefly introduce the P2P models under discussion. To facilitate comparison, we consider network topologies using a uniform graph-theoretic framework. We view peers as nodes in an undirected graph of size $\mathcal{M}$ where edges indicate connections between peers in the network. Each node N in the graph has, as an attribute, a routing table $\mathcal{T}_N = [e_1 : w_1, \ldots, e_k : w_k]$ that associates a weight w_i to each edge e_i ($1 \leqslant i \leqslant k$) incident on N. This represents the connections of node N to k neighbors in the graph. Unless otherwise stated, all weight values are equal in the graph.

2.1 Structured Models

As mentioned above, structured models enforce strict constraints on network evolution and resource placement. These constraints limit network robustness and node autonomy. Structured P2P models are good for building systems where controlled resource placement is a high priority, such as distributed file storage. However, they are not good models for systems with highly dynamic membership. The main advantage of these models is that the added constraints result in sublinear search mechanisms; each of these models has an associated native search mechanism that takes advantage of the added structure [20,24].

CAN. The Content Addressable Network (CAN), proposed by Ratnasamy et al. [20], is a framework for structured P2P systems based on a virtual d-dimensional Cartesian coordinate space on a d-torus. Nodes in a CAN graph have as an attribute the coordinates of a subspace of this space that are used in adding nodes and edges to the graph. Initially, the graph consists of one node and no edges. This initial node is assigned the entire virtual space. As nodes are added to the graph, they are assigned a subspace in the virtual space from a uniform distribution. The system self-organizes to adjust to a new node by

[4] http://www.gnutella.com

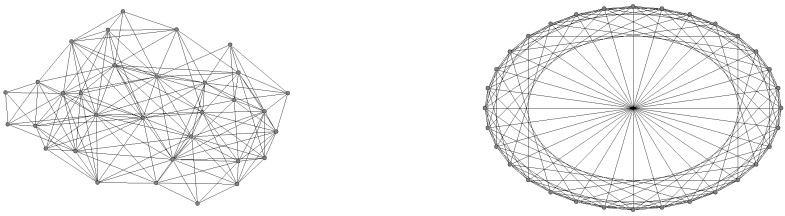

Fig. 1. 32 Node CAN and Chord Networks

adding edges from the new node to adjacent nodes in the space. A visualization of a 32 node CAN graph is given in Figure 1 on the left.[5]

Chord. Chord, proposed by Stoica et al. [24], is another self-organizing structured P2P system model. Nodes in a Chord graph have, as an additional attribute, a coordinate in a 1−dimensional virtual space (called a *ring*). When a new node N is added to the graph, the routing table attributes of the nodes adjacent to N on the ring are used to add edges between N and k other nodes distributed in the space. A visualization of a 32 node Chord graph is given in Figure 1 on the right.

2.2 Unstructured Models

Unstructured models strive for complete decentralization of decision making and computation. They require only local maintenance procedures and are topologically robust in the face of system evolution. These models are good for building highly dynamic systems where anonymity and minimal administrative overhead are prized.

Random Graph. We utilize the *Erdös-Rényi random graph* as a baseline model for comparison with unstructured networks [3,18]. There is one parameter in building a system with this topology: connection probability p. To build a random network based on this model, the graph initially has no edges. Then for each possible undirected edge between two distinct nodes in the graph, an edge is added with probability p.

PRU. The PRU (Pandurangan-Raghavan-Upfal) model for unstructured systems, proposed by Pandurangan et al. [19], is based on a simple network growth policy that ensures low graph diameter. In these graphs, nodes have a boolean attribute *inCache*, indicating their role in network evolution. The model has as parameters node degree K, minimum degree L, and maximum degree U. The graph starts with K nodes with attribute $inCache = True$. Each of these nodes has L edges incident on them from randomly chosen nodes within the group. When a new node N is introduced into the graph its *inCache* value is $False$,

[5] All graph visualizations in this paper were made with the Pajek package [6].

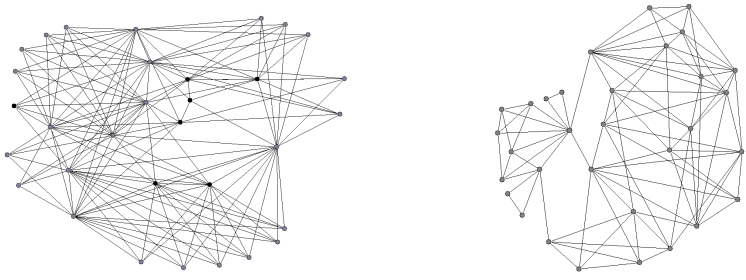

Fig. 2. 32 Node PRU and Hypergrid Networks

and edges are added between it and L randomly selected *inCache* nodes. If this addition causes any *inCache* node N_C to have more than U edges, N_C has its *inCache* value set to $False$, and a non-*inCache* node in the system is chosen to become *inCache* [19]. A visualization of a 32 node PRU graph is given in Figure 2 on the left with *inCache* nodes colored black.

Hypergrid. The Hypergrid model for P2P networks, proposed by Saffre and Ghanea-Hercock [21], builds a graph topology that enforces low graph diameter and bounded node degree. The graph grows as a simple k-ary tree with nodes on the leaf level of the tree having their $k-1$ free edges randomly connected to other nodes on the same level in the tree that have degree less than k. A visualization of a 32 node Hypergrid graph is given in Figure 2 on the right.

3 Unstructured P2P Search Algorithms

Search in a graph is defined as finding a path from a randomly chosen start node N_s to a randomly chosen destination node N_d. The *cost* of a search is the number of edges traversed in locating the destination node (i.e., the number of "messages" sent between peers in the network during the search process). There are two broad classes of search techniques for unstructured P2P graphs: *uninformed* (blind) and *informed* (heuristic) [25]. Uninformed algorithms utilize only local connectivity knowledge of the graph during search. Sometimes this is the best we can do; without the ability to maintain some local state, search can do little more than follow some systematic blind routine. If we can maintain some local state, then search can proceed in a more intelligent manner. In addition to basic connectivity, informed algorithms use some localized knowledge of the graph (such as "directional" metadata) to make heuristic decisions during search. In this section we consider two characteristic uninformed search algorithms, random Breadth-First-Search (BFS) [5,9,12] and k-random walk [1,15], and a generic informed search algorithm, GAPS [26].

3.1 Random Breadth-First-Search

Random BFS [5,9,12] is an uninformed search algorithm that has been proposed as an alternative to basic uninformed BFS ("flooding"). Basic BFS is a common

technique for searching graphs. Search begins at N_s by checking each neighbor for N_d. If this fails, each of these neighbors check their neighbors and this continues until N_d is found. The idea behind random BFS is to improve on the flooding method to reduce message overhead during search. This is attempted by randomly eliminating a fraction p of neighbors to check at each node. Search then proceeds from N_s with n_s neighboring nodes as follows: select $\lceil (1-p)n_s \rceil$ randomly chosen nodes adjacent to N_s, and return success if N_d is among them. Otherwise each of these neighbors randomly selects a $(1-p)$-subset of its neighbors. This process continues until N_d is located. If at any time during the search a node N contacts a "dead-end" node (a leaf in the graph), the search process backtracks to N and continues. It has recently been shown that there is an optimal value for p in certain restricted power-law networks [5].

3.2 *k*-Random Walk

Random walk on a graph is a well known uninformed search technique [1,15]. In this approach, a reduction in message overhead is attempted by having a single message routed through the network at random. Search proceeds from N_s as follows: randomly select one neighbor N. If $N \neq N_d$, then N similarly contacts one of its neighboring nodes, avoiding re-selecting N_s (if N has only one neighbor, it is forced to pass control back to N_s). This process continues until N_d is located. This search mechanism does not generate as much message traffic as the BFS algorithms since there is only one message being routed in the system. The trade-off is that the search response time is significantly longer. k-random walk extends this process to k random walkers that operate simultaneously with the goal of reducing user-perceived response time [15].

3.3 Generic Adaptive Probabilistic Search

As mentioned above, uninformed search is the best we can do lacking some local information. There have been several proposals to add "directional" metadata to uninformed search [4,12,26,29]. We consider here a simplification of these proposals which we call Generic Adaptive Probabilistic Search (GAPS), following the adaptive probabilistic search algorithm of Tsoumakos and Roussopoulos [26]. GAPS can be viewed as a minimally informed approach to searching in an unstructured system, making full use of the routing tables $\mathcal{T}_N = [e_1 : w_1, \ldots, e_k : w_k]$ associated with each node N. The weight w_i indicates the likelihood of successful search through neighbor N_i based on previous search results. Initially, $w_i = 1, \forall i$.

Search proceeds from N_s as follows: choose a single edge e_i from the routing table with probability $\frac{w_i}{\Sigma_{j=1}^{k} w_j}$, and return success if $N = N_d$ is adjacent on this edge. Otherwise, this neighbor selects one of its neighbors following the same procedure. When the destination node N_d is located, all nodes along the path from N_s to N_d (with loops removed) increment the weight in their neighbor tables for their successor in the path by 1. In this way, these nodes will be chosen with higher probability in future searches.

4 Simulation Results

To compare P2P network models in combination with search algorithms, we implemented them in a uniform framework. We considered using existing agent-based simulators [4,16], but decided that the level of implementation detail necessary for a clean investigation of topology/algorithm interaction necessitated a simple common framework. For each network of size $\mathcal{M}$ that we simulated, we used the following parameter values, which were chosen to build graphs of approximately equivalent edge count across all models:

- Random Graph: probability $p = \frac{2\mathcal{M}\log\mathcal{M}}{\mathcal{M}(\mathcal{M}-1)}$
- CAN: dimension $d = 3$
- Chord: edges $k = \log\mathcal{M}$
- PRU: *inCache* node count $K = \frac{\mathcal{M}}{4}$, lower bound $L = \log\mathcal{M}$, upper bound $U = 3L + 3$
- Hypergrid: degree $k = 2\log\mathcal{M} + c$, for constant $c < 6$.

Table 1. Statistics of Simulated Networks

Model	# Nodes	# Edges	Avg. Degree (min/max)	Avg. Distance	Diameter	Clustering Coefficient
Random	1024	10240	20.0 (7/34)	2.65	4	0.02
PRU	1024	10350	20.21 (10/34)	2.89	5	0.25
Hypergrid	1024	10239	20.0 (2/25)	3.71	5	0.124
CAN	1024	9524	18.60 (4/45)	4.85	10	0.50
Chord	1024	9728	19.0 (19/19)	3.45	5	0.16

4.1 Topological Properties

As briefly discussed in Section 1, P2P models and MAS are anticipated to grow small world networks that also possibly have power-law degree distributions

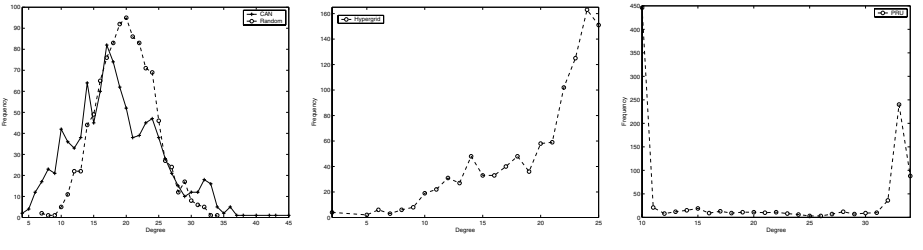

Fig. 3. Degree frequency distributions for CAN and Random model (left), HyperGrid model (center), and PRU model (right)

[3,5,10,11,18]. The results of our simulating the models under consideration for $\mathcal{M} = 1024$ are presented in Table 1. We measured these values using the Ucinet package [7]. Here, the *average distance* for a graph is the length of the shortest path between two nodes averaged over all node-pairs in the graph. The *diameter* of a graph is the length of the longest direct path in the graph between any two nodes. The *clustering coefficient* of a graph is the proportion (averaged over all nodes) of nodes adjacent to a particular node that are also adjacent to each other [28]. The node degree frequencies for the models are plotted in Figure 3.

4.2 Search Performance

We now describe our experimental setup for measuring search performance. We were interested in the actual number of edges traversed to find a node in the system. The studies discussed in Section 1.1 have primarily considered the *probability* of successful search. We were looking at the *cost* of 100% success for each

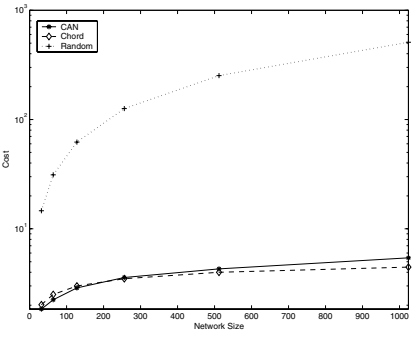

Fig. 4. Search performance comparison of structured models (CAN, Chord) using their native search algorithms against an unstructured model (Random) using BFS

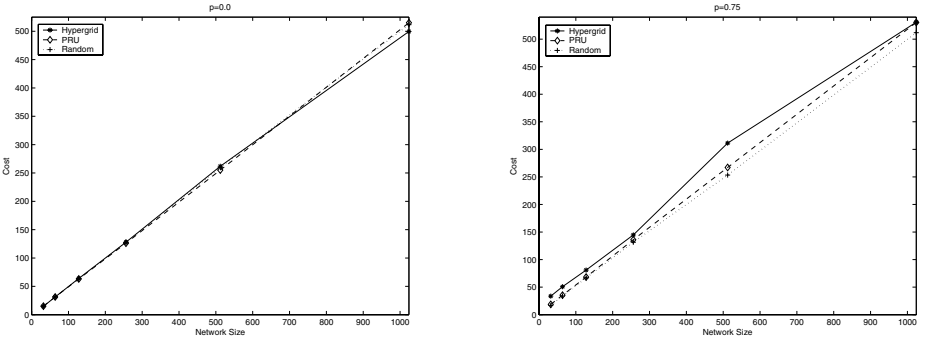

Fig. 5. Random BFS search performance across Hypergrid, PRU and Random models. Cutoff probability = 0.0 (left) and 0.75 (right).

search (i.e., Time To Live, TTL $= \infty$). We measured search cost, on simulations of network size 2^n for $5 \leqslant n \leqslant 10$, as the average of 5000 searches on each size (specifically: 50 simulated networks, 100 searches on each, for all 6 network sizes). For measurements of the GAPS algorithm, we "weighted" some fraction P of nodes in the system more heavily (i.e., P% of the nodes are "popular") to be the destination for some fraction W of the searches. We skewed search in this manner since the general efficacy of GAPS is dependent upon there being popular nodes in the system that are the destination nodes for a higher than average proportion of the searches. We also "primed" the network with 100 messages before measuring GAPS cost so that we could distinguish its behavior from random walk. The results of our simulations are presented in Figures 4 – 9.

5 Discussion

As mentioned above, the defining characteristics of a small-world network are low diameter and high clustering coefficient [28]. The values in Table 1 clearly indicate that all of the models (except the random model) grow small-world topologies. Chord, with a constant degree distribution, does not exhibit a power law. None of the degree distributions plotted in Figure 3 follow power-laws: CAN (left) follows a Poisson distribution (like the random graph) because it is built by assigning nodes in the graph using a uniform hash function [20]. In the case of Hypergrid graphs (center), the bulk of the nodes have maximum degree while some linearly decreasing number of nodes at the leaf level fail to establish maximum degree. PRU (right) has a highly skewed distribution: the "bump" at degree 10 represents the lower bound L on degree, while the peak at degree 33 represents nodes that have reached the upper bound U on degree. There are a nontrivial number of nodes with degree 34. These nodes were allowed to have $U + 1$ neighbors to handle an error condition in the PRU growth protocol [19]. The few intermediate nodes with degree between L and U are those currently *inCache*.

Turning to performance, Figure 4 illustrates the value of structure: the CAN and Chord native search mechanisms give $O(\log \mathcal{M})$ search performance. The cost of BFS on random graphs (typical of the unstructured models) increases linearly with network size $\mathcal{M}$, with cost roughly $\mathcal{M}/2$. Clearly, the native search mechanisms of structured networks outperform, by several orders of magnitude, flooding search on unstructured networks.

Next, we compare the three search algorithms for unstructured networks. The results for BFS with 0.0 and 0.75 cutoff values is given in Figure 5, for 1 and 16-random walk in Figure 6, and for GAPS, with 5% of the nodes popular receiving 75% of search requests, in Figure 7 (left). Clearly, all variants of random BFS have the same cost (indicating that randomness does not enhance basic BFS) and have lower cost than both GAPS and k-Random Walk. Also, GAPS has lower cost than the Random Walk search algorithm. The long term performance improvement of GAPS algorithm for the Random Graph model is presented in Figure 7 (right). Clearly this algorithm improves over time (albeit at a very

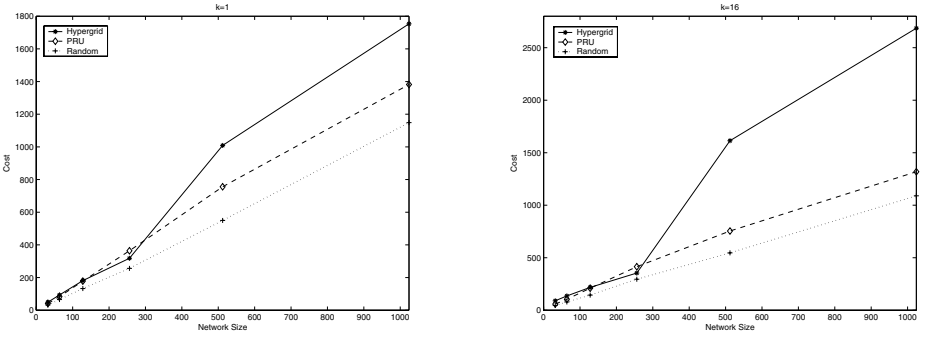

Fig. 6. k-Random Walk search performance across Hypergrid, PRU and Random models. k = 1 (left) and 16 (right).

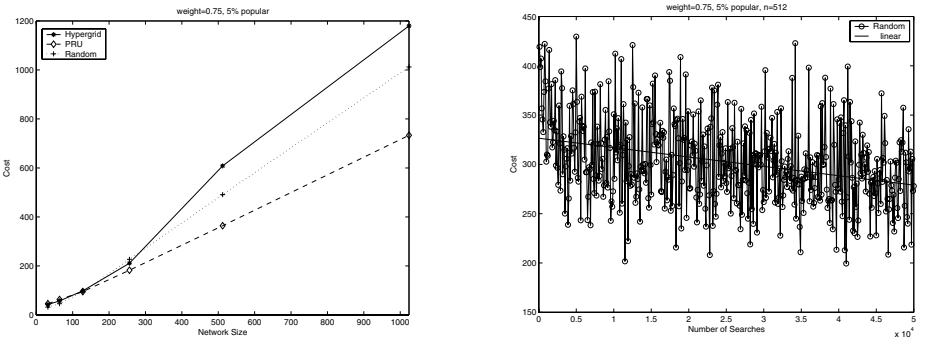

Fig. 7. GAPS (weight = 0.75, popularity = 5%) search performance (left). GAPS search performance over time, Random graph (right).

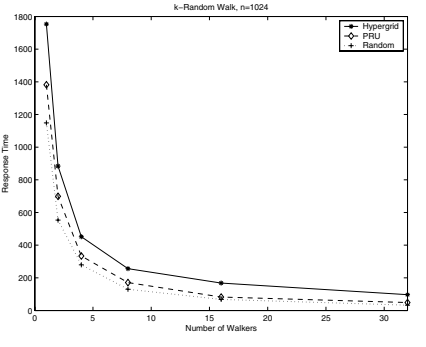

Fig. 8. k-Random Walk normalized cost (User Response Time = Cost/Number of Walkers

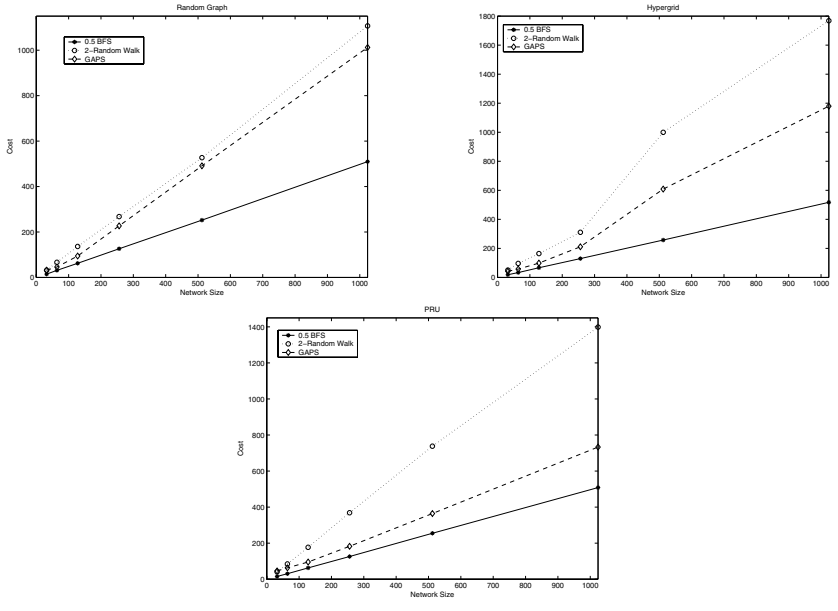

Fig. 9. Performance of search algorithms (BFS, Random Walk and GAPS) across random model (top left), Hypergrid (top right) and PRU model (bottom)

gradual rate). We also compare the user-perceived response time (that is, the normalized cost of search) of all three P2P models for k-Random Walk ($k = 1, 2, 4, 8, 16, 32$) in Figure 8. Normalized cost improvement is equivalent across all three models.

Finally, we independently consider search performance on each of the three topologies. From Figure 9, it is evident that the random graph scales well for all the search algorithms. Hypergrid has similar search cost as that of PRU and Random graph for small size networks but as the network size increases, its performance degrades. Random Walk involves the highest cost in all three graphs, making GAPS a good alternative to $k-$random walk. Overall, these experiments clearly indicate that the random graph model and BFS requires lowest cost for unstructured networks.

6 P2P Models, Search Algorithms and Learning Modules

The P2P models and search algorithms discussed and compared in this paper have recently been re-implemented in Java and integrated into the IVC Software Framework in the InfoVis Cyberinfrastructure under development in the School of Library and Information Science at Indiana University.[6] The IVC Software Framework enables non-programmer users to run diverse data mining, modeling and visualization algorithms in a menu driven way.

[6] http://iv.slis.indiana.edu/

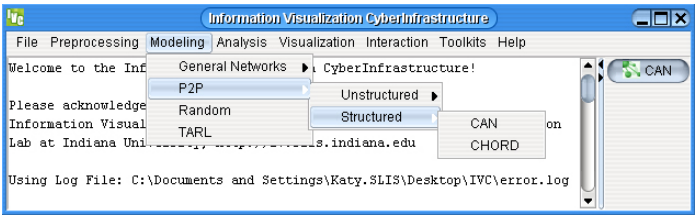

Fig. 10. Main application window of the IVC Software Framework

A snapshot of the interface to the IVC Software Framework is given in Figure 10. Continuous feedback on user requests and algorithmic results is printed in the background of the main application window. Generated networks can be analyzed using the Network Analysis Toolkit available under the 'Toolkits' menu or by running one of the diverse search algorithms under the 'Analysis' menu. Networks can be visualized using algorithms available under the 'Visualization' menu. All algorithms in the IVC Software Framework are extensively documented online. In addition, two Learning Modules are available online that aim to educate about the Error and Attack Tolerance of Networks and about the Search Performance of P2P Networks.

7 Conclusions and Future Work

In this paper we explored the topological properties and search performance of structured and unstructured P2P models using simulations of the CAN, Chord, Hypergrid, and PRU models and the random BFS, k-random walker, and GAPS search algorithms. Our goal was to provide a basis for a better understanding of the role of topology in search performance and to highlight the strengths and weaknesses of these models and algorithms.

We discovered that most of these models do indeed grow as small worlds with low diameter and high clustering coefficients. None of the models developed power-law degree distributions. We also found that basic BFS overall had lowest search cost across all unstructured models and that the random graph topology supports the lowest cost search overall using BFS. Furthermore, we determined that random cutoff does not improve the cost of BFS. We also found that increasing the number of walkers in random walk does not improve search cost; in fact, this just trades network load for user perceived response time. Finally, we found that the GAPS algorithm performs well as an alternative to k-random walk on all networks. These results indicates the need to study more closely algorithms that intelligently adapt to system dynamism and usage.

The next step in this research is to undertake a complete formal investigation of the GAPS algorithm as a paradigmatic informed search algorithm. Its generality and simplicity may give a good handle on designing efficient informed search algorithms for small-world graphs that outperform BFS. Another important step is to investigate unstructured topologies to specifically support GAPS.

Finally, an investigation of recent results which have applied percolation theory to the problem of search in power-law graphs [5,22] can profitably be pursued in our simulation framework.

Acknowledgments. We thank Beth Plale, Cathy Wyss, the reviewers, the Indiana University Database Group, the AP2PC 2004 workshop participants, and Gopal Pandurangan for their feedback and discussions on this paper. This work is supported by a National Science Foundation CAREER Grant under IIS-0238261 to the third author.

References

1. Adamic, Lada, Rajan Lukose, Amit Puniyani, and Bernardo Huberman. Search in Power-Law Networks. *Physical Review E*, 64(4):46135-46143, 2001.
2. Akavipat, Ruj, Le-Shin Wu and Filippo Menczer. Small World Peer Networks in Distributed Web Search. *Proc. ACM WWW2004*, pp. 396-397, 2004.
3. Albert, Réka and Albert-László Barabási. Statistical Mechanics of Complex Networks. *Reviews of Modern Physics*, 74(1):47-97, 2002.
4. Babaoğlu, Ö., H. Meling, and A. Montresor. Anthill: A Framework for the Development of Agent-Based Peer-to-Peer Systems. *Proc. IEEE ICDCS'02*, pp. 15-22, 2002.
5. Banaei-Kashani, Farnoush and Cyrus Shahabi. Criticality-based Analysis and Design of Unstructured Peer-to-Peer Networks as "Complex Systems." *Proc. IEEE/ACM CCGRID'03*, pp. 351-358, 2003.
6. Batagelj, Vladimir and Andrej Mrvar. Pajek: Package for Large Network Analysis. http://vlado.fmf.uni-lj.si/pub/networks/pajek/
7. Borgatti, S.P., M.G. Everett, and L.C. Freeman. Ucinet for Windows: Software for Social Network Analysis. Harvard: Analytic Technologies, 2002.
8. Decker, K., K. Sycara, and M. Williamson. Middle-Agents for the Internet. *Proc. IJCAI97*, pp. 578-583, 1997.
9. Dimakopoulos, Vassilios V. and Evaggelia Pitoura. A Peer-to-Peer Approach to Resource Discovery in Multi-Agent Systems. *Proc. CIA 2003*, Springer LNCS 2782, pp. 62-77, 2003.
10. Faloutsos, M., P. Faloutsos, and C. Faloutsos. On Power-Law Relationships of the Internet Topology. *Proc. ACM SIGCOMM*, pp. 251-262, 1999.
11. Jovanović, M., F. Annexstein, and K. Berman. Modeling Peer-to-Peer Network Topologies Through "Small-World" Models and Power Laws. *IX Telecommunications Forum TELFOR 2001.*
12. Kalogeraki, Vana, Dimitrios Gunopulos and D. Zeinalipour-Yazti. A Local Search Mechanism for Peer-to-Peer Networks. *Proc. ACM CIKM'02*, pp. 300-307, November 2002.
13. Kleinberg, Jon. Navigation in a Small World. *Nature*, 406:845, August 2000.
14. Koubarakis, Manolis. Multi-Agent Systems and Peer-to-Peer Computing: Methods, Systems and Challenges. *Proc. CIA 2003*, Springer LNCS 2782, pp. 46-61, 2003.
15. Lv, Qin et al. Search and Replication in Unstructured Peer-to-Peer Networks. *Proc. ACM ICS'02*, pp. 84-95, 2002.
16. Minar, N., R. Burkhart, C. Langton, and M. Askenazi. The Swarm Simulation System, A Toolkit for Building Multi-Agent Simulations. *Technical Report, Swarm Development Group*, June 1996.

17. Milojičić, Dejan S., et al. Peer-to-Peer Computing. *HP Labs Technical Report HPL-2002-57*, 2002.
18. Newman, M.E.J. The Structure and Function of Complex Networks. *SIAM Review*, 45(2):167-256, 2003.
19. Pandurangan, G., Prabhakar Raghavan, and Eli Upfal. Building Low-Diameter Peer-to-Peer Networks. *IEEE J. Select. Areas Commun.*, 21(6):995-1002, August 2003.
20. Ratnasamy, Sylvia et al. A Scalable Content-Addressable Network. *Proc. ACM SIGCOMM*, pp. 161-172, August 2001.
21. Saffre, Fabrice and Robert Ghanea-Hercock. Beyond Anarchy: Self Organized Topology for Peer-to-Peer Networks. *Complexity*, 9(2):49-53, 2003.
22. Sarshar, Nima, P. Oscar Boykin, and Vwani Roychowdhury. Percolation Search in Power Law Networks: Making Unstructured Peer-to-Peer Networks Scalable. *Proc. IEEE P2P2004*, pp. 2-9, 2004.
23. Shehory, O. A Scalable Agent Location Mechanism. *Proc. ATAL'99 Intelligent Agents VI*, pp. 162-172, 1999.
24. Stoica, Ion et al. Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications. *IEEE/ACM Trans. on Networking*, 11(1): 17-32, February 2003.
25. Tsoumakos, Dimitrios and Nick Roussopoulos. A Comparison of Peer-to-Peer Search Methods. *Proc. ACM WebDB 2003*, pp. 61-66, 2003.
26. Tsoumakos, Dimitrios and Nick Roussopoulos. Adaptive Probabilistic Search for Peer-to-Peer Networks. *Proc. IEEE P2P2003*, pp. 102-109, 2003.
27. Walsh, Toby. Search in a Small World. *Proc. IJCAI99*, pp. 1172-1177, July-August 1999.
28. Watts, Duncan and Steven Strogatz. Collective Dynamics of 'Small-World' Networks. *Nature*, 393:440-442, June 1998.
29. Yang, Beverly and Hector Garcia-Molina. Improving Search in Peer-to-Peer Networks. *Proc. IEEE ICDCS'02*, pp. 5-14, 2002.